VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2020 Bc. Adam Kovář

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

BEZPEČNÁ IMPLEMENTACE TECHNOLOGIE BLOCKCHAIN
SECURE IMPLEMENTATION OF BLOCKCHAIN TECHNOLOGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Adam Kovář**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Lukáš Malina, Ph.D.**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Adam Kovář ***ID:*** 186119
Ročník: 2 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Bezpečná implementace technologie blockchain

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s technologií blockchain. Analyzujete bezpečnostní a technologické možnosti této technologie. Získané poznatky využijte pro aplikaci na SAP Cloud Platform pro modul Treasury and Risk Management. V rámci práce vytvořte modelový příklad pro zápis a čtení z blockchainu na výše zmíněné platformě.
V rámci diplomové práce vytvořte aplikaci pro využití v modulu Treasury and Risk Management, který bude ukládat data o transakcích do blockchainu a následně bude potvrzováno různými entitami připojených do privátního blockchainu.

DOPORUČENÁ LITERATURA:

[1] DRESCHER, Daniel. Blockchain basics. Berkeley, California: Apress, 2017. ISBN 148422603.

[2] GUPTA, Raja, Nagesh CAPARTHY, Vijayalakshmi GOPALAKRISHNA a Atul LADIA. Introducing Blockchain with SAP Leonardo [online]. Boston (MA): Rheinwerk Publishing, 2019 [cit. 2019-09-27]. ISBN 978-1-4932-1-08-0. Dostupné z: https://www.sap-press.com/introducing-blockchain-with-sap-leonardo_4843/

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 1.6.2020

Vedoucí práce: doc. Ing. Lukáš Malina, Ph.D.
Konzultant: Ing. Rudolf Bryša PhD. (SAP Labs Czech republic)

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato práce popisuje základní problematiku a implementaci blockchain technologie pro SAP Cloud platform s důrazem na bezpečnost citlivých dat. Tato diplomová práce implementuje správu celého procesu bankovního akreditivu. Práce popisuje stavbu blockchainu od elementárních stavebních bloků až po samotné napojení na SAP Cloud Platform. Tato implementace je zaměřena na výběr všech funkčních bloků s nahlédnutím na možné alternativy pro řešení sledování průběhu bankovního akreditivu za účel zajištění integrity a bezpečnosti dat. Možné alternativní řešení jsou popsána a je zde nastíněno pro jaké implementace je právě alternativní řešení vhodné. V rámci diplomové práce je vypracováno jádro programu, které následně může sloužit pro vývoj produktivní aplikace, která bude sloužit pro správu bankovního akreditivu. Implementace obsahuje všechny potřebné bloky pro fungování a sledování průběhu bankovního akreditivu.

KLÍČOVÁ SLOVA

blockchain, zabezpečení, kryptoměny, treasury, risk management, sap, sap cloud platform, šifrování, hashovaní, Merkel tree, multichain, hyperleger fabric, konsenzuální algoritmy, sap leonardo, bankovní akreditiv

ABSTRACT

This thesis describes basis of blockchain technology implementation for SAP Cloud platform with emphasis to security and safety of critical data which are stored in blockchain. This diploma thesis implements letter of credit to see and control business process administration. It also compares all the possible technology modification. Thesis describes all elementary parts of software which are necessary to implement while storing data and secure integrity. This thesis also leverages ideal configuration of each programable block in implementation. Alternative configurations of possible solutions are described with pros and cons as well. Another part of diploma thesis is actual working implementation as a proof of concept to cover letter of credit. All parts of code are design to be stand alone to provide working concept for possible implementation and can source as a help to write productive code. User using this concept will be able to see whole process and create new statutes for whole letter of credit business process.

KEYWORDS

blockchain, security, cryptocurrencies, treasury, risk management, sap, sap cloud platform, encryption, hashing, merkel tree, multichain, hyperleger fabric, consensus algorithms, sap leonardo, letter of credit

KOVÁŘ, Adam. *Bezpečná implementace technologie blockchain*. Brno, 2019, 65 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Ing. Lukáš Malina, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Bezpečná implementace technologie blockchain“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Děkuji panu doc. Ing. Lukáši Malinovi Ph.D. cenné rady a pomoc při vedení diplomové práce. Mé poděkováni patři též Ing. Bc. Rudolfu Bryšovi Ph.D. za odborné vedení, podnětné návrhy k práci a za spolupráci při získávání údajů pro výzkumnou část práce.

Obsah

1 Úvod

Cílem této diplomové práce je vysvětlení a uvedení do problematiky blockchain technologie a bezpečné implementace na příklad pro bankovní akreditiv. Blockchain struktura je v práci popsaná od elementárních bloků až po různé možné implementace konsenzuálních algoritmů pro ověřování dat. Následně je také shrnutý popis pro implementaci na platformě SAP Cloud Platform.

V teoretické části je popsán blockchain jako stavební blok dat, která jsou šifrovaná, a způsob přístupu k jednotlivým datovým jednotkám. Bude zde shrnuto hashování dat, možnosti využití blockchainu, kryptoměny a také jiné implementace blockchain technologie. Součástí práce pak bude i popis struktury Merklova stromu jako základního prvku pro uložení všech dat do blockchainu. Způsob uložení bude následně popsán i na využití v kryptoměnách a bude také vysvětleno, proč nám poskytuje velmi vysokou bezpečnost v případě útoku. Další část se věnuje využití konsensuálních algoritmů pro ověřování a rozdíly mezi nimi. Dále je zde popsán princip bankovního akreditivu, pro který bude následně v rámci praktické části vytvořena implementace. Následující část pak už bude zaměřena na popis SAP Cloud Platform a možnosti aplikace různých technologií blockchainu na této platformě.

V praktické části této práce bude popsáno, jaké máme možnosti implementace pro vlastní řešení do blockchain prostředí právě pro SAP Cloud Platform. Praktická část a nápad pro implementaci vznikl ve firmě SAP jako součást interních inovací firmy a tato implementace slouží jako koncept pro možný vývoj finálního řešení. Součástí diplomové práce bude vytvoření aplikace, jež bude mít na starost korespondenci mezi entitami, které se budou účastnit bankovního akreditivu. Součástí aplikace bude také návrh a popis všech stavebních bloků, nutných k zprovoznění funkčnosti této implementace. Součástí řešení bude také úvaha nad výběrem jednotlivých technologií a způsobu přístupu, aby šlo o co nejvíce bezpečné a efektivní řešení. Výsledná aplikace pak dokáže zapisovat a informovat účastníky bankovního akreditivu o stavu a o tom, jaké kroky jsou nutné pro vykonání celého procesu. Závěrem pak bude shrnutí a vyhodnocení kvality implementace a zvážení možných alternativ při řešení.

Blockchain technologie zaznamenala v posledních letech velké rozšíření pro využití funkcionality kryptoměn. Paralelně vznikají, ale i jiné inovativní nápady, kde lze tato technologie implementovat. Motivace pro implementaci blockchain technologie pro bankovní akreditiv je založená na nutnosti obchodního styku s méně důvěryhodnou protistranou. Součástí bankovního akreditivu je žádoucí kontrola plnění závazků. Blockchain tedy jako decentralizovaná databáze, která umožňuje pouze přidávat a číst uložené záznamy, poskytuje ideální substituci aktuálně využívané komunikace pro bankovní akreditiv.

2 Technologie blockchain

Blockchain jako technologie je druh distribuované decentralizované databáze, která neustále rozšiřuje svůj záznamový chronologicky uspořádaný řetězec. Data uvnitř jsou zabezpečena asymetrickou kryptografií proti falzifikaci.

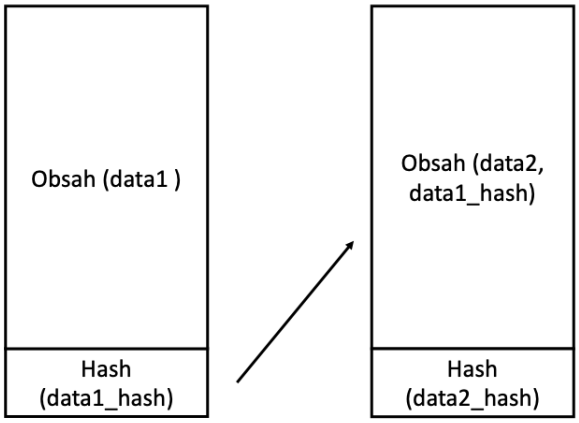
Na rozdíl od klasické databáze, která bývá nejčastěji spravována jednou centrální entitou, právě decentralizace eliminuje nutnost existence centrálního správce. Následně se pak dá zapisovat do databáze pouze na základě předem definovaných pravidel pro každého účastníka. Následné ověřování transakcí pak už závisí na odpovědnosti samotných účastníků v blockchainu tzv. peerů. Tyto technologie tedy najdou využití ve vysoce automatizovaných procesech a urychlí samostatné interní manuální procesy, následně i sníží jejich cenu díky eliminaci centrálních entit, které takto vytvořené databáze dříve spravovaly, více informací viz [1].

2.1 Účastníci a databáze

Účastník nebo také peer v síti blockchain představuje jednoho účastníka, který přistupuje do blockchainu a je připojený pomocí sítě do databáze (blockchain řetězce). Každý jeden účastník musí splňovat bezpečností pravidla pro připojení do takto vytvořené databáze, což je tedy tabulka umístěná na severu, kam se vkládají data neustále rostoucím klíčem pro zajištění unikátnosti jednotlivých záznamů. Do takto vzniklé datové sítě s tabulkou připravenou nejčastěji třetí stranou následně posíláme šifrované datové buňky. Jejich princip fungování bude popsán v dalších kapitolách mnohem podrobněji, více informací viz [1, 2].

2.2 Datová buňka

Datová buňka v blockchain síti se skládá ze dvou bloků. První z nich jsou samotná transakční data, která chceme do databáze uložit. Nejčastěji jde o data uložená v předem definované struktuře nebo formátu, ať už ve formě standardizovaného SWIFT formátu nebo, XML formátu. Na konci buňky je pak část hash, která je v následujícím bloku asymetricky ověřena a vložena do pole obsahu, čímž je pak tedy zajištěna integrita a bezpečnost takto dvou po sobě vložených datových buněk, jako vidíme na obrázku níže. Nemusí to být vždy pouze takto jednoduché, dost často se využívá kombinace napojení datových buněk do struktur, např. Merkle tree, která zvyšuje bezpečnost. V datové buňce pak vzniká další záznam hash hodnoty z jiného datového pole pro zajištění konzistence struktury, ale narůstá paměťová náročnost a to může být nežádoucí, více informací viz [1].



Obr. 2.1: Jednoduché datové buňky

2.3 Struktura Merkle tree

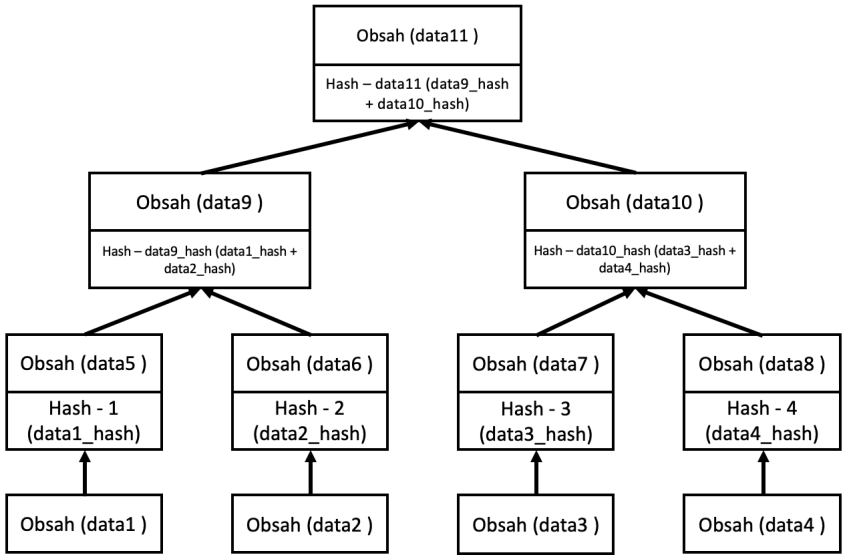
Hašovací strom neboli Merkle tree je zajímavá datová struktura, která je schopna uložit data do technologie blockchain velmi bezpečným způsobem. Hašovací strom spojuje všechny transakce do jedné hashové hodnoty, která je navenek pro uživatele zobrazena. Všechny ostatní hodnoty, které obsahují jednotlivé bloky, jsou pak uživateli neznámé. Jde o klasickou strukturu binárního stromu, kde máme vždy kořenový uzel s hash hodnotou odvozenou z předchozích bloků. Strom je vždy vyvážený a jednotlivé bloky na sebe vždy odkazují směrem zespodu nahoru.

Každý z bloků v této technologii má hash hodnotu z předchozí transakce, proto je zde velmi rychlá kontrola, zda byla transakce přidána, protože se vždy mění hodnota kořenové transakce v hlavičce. V případě, že je potřeba dohledat záznam o transakci, není nutno stahovat celý strom, ale vždy se stáhne jen ta cesta, které je potřeba, a dosáhneme tedy velmi rychlé odpovědi. Takto implementovaná technologie je stavební kámen ve známých kryptoměnách, jako jsou Bitcoin nebo Ethereum, více informací viz [2, 3].

Vytvoření kořenového záznamu

Hašovací strom je klíčovou součástí této technologie. Je tedy důležité vědět, jak je vytvořen samotný kořenový záznam uvnitř. Základem pro vytvoření kořenového ID je spočítání všech bloků, které jsou zaznamenány do databáze. Při vytváření stromu po přidání nové transakce dojde k rozdělení na páry v chronologickém rozložení a poté vytvoříme vyvážený binární strom, kde následně z každého záznamu

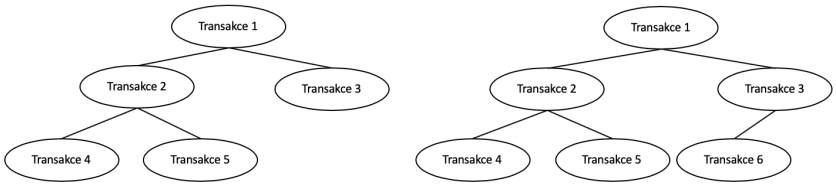
vytvoříme odpovídající hash hodnotu. Tímto způsobem bude tedy každý blok reprezentovat hash hodnota. Následně začneme od nejnižší vrstvy stromu kombinovat tyto Hash ID. Takto zkompletujeme celý binární strom do jedné ID hodnoty, která se bude následně označovat jako kořenové ID. Takhle zajistíme, že všechny entity připojení do této databáze budou mít vždy stejnou a aktuální verzi dat. Při jakémkoliv pokusu o změnu dojde tedy k přepsání hodnoty, následná detekce podvodu bude velmi rychlá, protože se objeví už v jiné hodnotě v poli ID u kořenového záznamu. Pro lepší představu o systému Merkle tree je přiložen obrázek níže, více informací viz [3].



Obr. 2.2: Datová struktura Merkle tree

Přidání nového bloku do struktury Merkle tree

Přidání nového bloku je znázorněno na obrázku níže. Jde o přidávání transakcí s požadavkem na udržení struktury vyváženého binárního stromu s tím, že vždy po přidání nového prvku je přepočítán úvodní transakční blok, v tomto případě č. 1.



Obr. 2.3: Přidání transakce do Merkle tree

Zvýšená ochrana dat

V některých obchodních procesech je velmi důležité, aby při přenosu dat nebyla některá citlivá data viditelná třetím stranám, které mají kontrolu nad potvrzováním transakcí nebo musí tyto transakce nějak schválit. Tato data jsme schopni skrýt v případě právě využití technologie hašovacího stromu, který zpřístupní vždy jen tu část stromu, kde máme zájem data číst, nebo zapisovat, více informací viz [2].

2.4 Práce s daty v technologii blockchain

Hlavním cílem blockchainu je decentralizace dat, bezpečné uložení a také přístup k nim. V této kapitole je popsáno, jak jsou data v blockchainu šifrována a jakým způsobem k nim můžeme přistupovat, více informací viz [1, 2].

Šifrování dat

Jedna z klíčových vlastností blockchainu je ochrana dat proti falzifikaci. Většina blockchain aplikací využívá asymetrické kryptografie, kde se využívá podepisování privátním klíčem a následné potvrzení pravosti transakce odeslané do blockchainu. Jeden z limitů pro využití veřejného blockchainu je obecné nařízení o ochraně osobních údajů GDPR, které nás limituje, jaké data mohou být dostupná k zobrazení všem účastníkům blockchain sítě. V této práci je tedy řešením soukromý blockchain, který bude zpřístupňovat a vycházet z předem podepsané smlouvy mezi účastníky, kde budou definována pravidla a sankce v případě porušení pravidel nebo úniku citlivých dat, více informací viz [1, 2].

Hashování dat

Samotná data jsou chráněna asymetrickou kryptografií s podporou konsenzuálního algoritmu pro ověření integrity a důvěrnosti dat. To ale nestačí, a proto budou uložena ve struktuře Merkle tree, která využívá několikastupňové hashování dat pro zajištění maximálního utajení při jejich přenášení. Hashovací funkce je funkce, při které námi vytvořený datový řetězec překonvertujeme vždy na stejně dlouhý datový řetězec, a následný přesun dat je tedy bezpečný pro přenos, jelikož nikdo bez klíče není schopný takto šifrovaná data při přenosu dešifrovat. V samotném Merkle tree se využívá mnohonásobné hashování s odkazy na předchozí hashe. Toto bude více popsáno v samostatné kapitole, více informací viz [1, 2].

Práce s daty

Práce s daty v blockchain technologii je možná pouze na základě ověřovacích pravidel pro přístup do aplikace a podpisu smlouvy, v našem případě s definovaným privátním blockchainem. Tímto způsobem dostaneme přístup do prostoru, kde můžeme vkládat pouze další nové transakce, ale přepisování a mazání dříve vložených není možné. Tato vlastnost zajistí dostatečnou míru nepopiratelnosti dat, jelikož je to základní vlastnost blockchain struktury. Úpravy jsme ale schopni adresovat na předešlé bloky a měnit data na aplikační vrstvě, což je velmi důležitá vlastnost pro uživatele, více informací viz [1, 2].

3 Typy blockchain technologií

Blockchain se dá použít na mnoho problémů, ale jen některé aplikace využijí všech jeho výhodných vlastností a budou ekonomicky výhodné. Před zahájením softwarového vývoje by se mělo diskutovat o tom, jaký bude zvolen, protože typy blockchainu se liší v mnoha pohledech na použité technologie uvnitř. To stejné se však dá říct i o typech blockchainu, které můžeme takto vytvořit na SAP Cloud Platform. Pro každý problém se bude vždy hodit jiný. Liší se i ve složitosti implementace jednotlivé technologie, více informací viz [2, 6].

3.1 Typy blockchainu

Nejpoužívanější rozdělení blockchain implementací je jejich otevřenost vůči potenciálním účastníkům a jejich přístupu do sítě blockchain. Použití pro kryptoměny nebo dodavatelské řetězce určitě vyžadují jinou implementaci z podhledu nutné aplikace pro zajištění sdílené databáze mezi účastníky za účelem urychlení správy dat, více informací viz [6].

Veřejný blockchain

Veřejný blockchain nejčastěji najdeme v kryptoměnách, kde je přítomnost a následné poskytnutí výpočetního výkonu pro umožnění komunikace v síti. Proto je zájem nových účastníků v síti finančně ohodnocen, více informací viz. [6].

V ideálním veřejném blockchainu bude existovat nekonečně mnoho účastníků, kteří budou neustále potvrzovat transakce jiných, a čím více jich, bude tím rychleji může samotná síť fungovat. Díky tomu není potřeba pro jedno ověření čekat na odpověď mnoha vzdálených účastníků, ale bude stačit 51 % účastníků s co nejmenší odezvou. S tím ovšem přichází i riziko, že takto veřejný blockchain bude eventuálně napaden hackerským útokem a můžou být odcizena data nebo kryptoměna, která oceňuje výpočetní stroje. Následně klesne důvěra v technologii aplikace. To může mít vliv na cenu měny, kde následně hrozí kolaps měny nebo sítě, když mnoho lidí odejde například z toho důvodu, že měna není dostatečně bezpečná pro uložení jejich peněz, více informací viz [6].

Samostatný veřejný blockchain může být ovšem jak otevřený, tak uzavřený. Otevřený veřejný blockchain jsou již výše zmíněné kryptoměny. Uzavřená je pak například aplikace pro volby, které by se ze zastaralého lístkového volení mohl velmi jednoduše přesunout do blockchainu a zajistit absolutní integritu i bezpečnost. V rámci této aplikace bude mít vždy každý občan přístup pouze jednou a bude identifikován jednoznačně přiřazeným ID, které zajistí, aby nemohlo vzniknout více záznamů pro

jednoho účastníka například identifikací pomocí rodného čísla, více informací viz [6].

Privátní blockchain

Privátní blockchain najde využití právě ve většině ekonomických scénářů, protože jde jednotlivým stranám přidělit určitou míru důvěry a sankcí v případě porušení dohod mezi účastníky.

Od veřejného blockchainu se liší právě omezením toho, kdo může do blockchainu vstoupit. Ke vstupu bude nutno vždy definovat primární zabezpečení a přístupové udaje, ale i sankce při porušení dohody o vedení nesprávných dat.

Stejně jako veřejný blockchain se i ten privátní dělí na otevřený a uzavřený. Do kategorie otevřených spadají všechny aplikace, které budou spravovat např. dodavatelské řetězce pro zajištění kvality materiálů a k lepší správě skladů. Další možností využití jsou také státní finanční operace, aby se dokázalo zabránit zpronevěření financí z důvodu lehce dohledatelné cesty peněz zpět. Uzavřený blockchain pak reprezentuje hlavně data, která se budou ukládat na bezpečné místo. Tuto vlastnost nejvíce ocení například daňová přiznání pro jednotlivé subjekty, kde je nežádoucí, aby ho kdokoliv jiný viděl, více informací viz [6, 7].

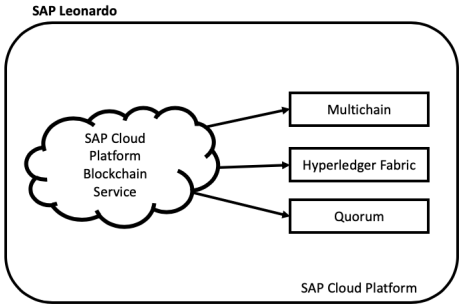
Konsorcium

Vždy půjde o nějakou skupinu entit, které budou mít zájem vidět do sdílené databáze z důvodu kontroly a ověřování transakcí mimo jejich infrastrukturu, například pro konsorcium R3, které sjednocuje 43 soupeřících bankovních domů. Toto řešení pak bude více rozebráno právě v rámci řešení diplomové práce, více informací viz [6, 7].

3.2 Platforma SAP Blockchain Applications and Services

SAP Blockchain Applications and Services je platforma, která nám poskytuje velmi jednoduchým způsobem možnost aplikovat technologie blockchain s námi zvolenými parametry na databázi. Samotná BaaS patří pod SAP Leonardo. Jde o uskupení technologií sloužících k digitální transformaci dat v parametrech pro 21. století. Tyto open source služby nám pak slouží k bezpečnému propojení jednotlivých účastníků dle námi zvolených parametrů pro síť.

SAP Cloud Platform Blockchain service není nový druh blockchain technologie, ale umožní nám aplikovat naše specifické parametry pro již asistující technologie, jako jsou Hyperledger Fabric, MultiChain nebo Quorum. Implementace probíhá v několika krocích a je rozdělena do mnoha úrovní, aby bylo vše přehledně strukturalizováno, více informací viz [2].



Obr. 3.1: Blockchain services

Platforma MultiChain

MultiChain je open-source platforma pro vytváření privátních blockchain sítí. Mezi hlavní výhody patří, že dokáže nad námi definovanou sítí provádět až 1000 transakcí za sekundu. Jde o méně zabezpečený model, je tedy velmi důležité, aby účastnící a jejich akce v síti byli vždy kontrolováni. MultiChain ovšem na druhou stranu nabízí velmi jednoduchou instalaci, a jde tedy spíše o testovací prostředí před tím, než aplikujeme blockchain v plné šíři na vybraný problém. Každý účastník se vždy jen připojí s API kódem a může do této sítě hned přidávat a číst bloky s vysokou přístupovou rychlostí, protěže tato síť neobsahuje žádné doplňky, jako jsou modifikace nad bloky.

MultiChain je příliš jednoduchá technologie a pro naši aplikaci v rámci praktické části není dost flexibilní a nebude tedy zvolena, více informací viz. [2, 7].

Platforma Hyperledger Fabric

Hyperledger Fabric je mnohem komplexnější technologie. Nabízí nám vytvoření více databází/ hlavních knih v jedné síti. Poskytuje nám také mnohem větší spektrum bezpečnosti, modularity a integrity. Pro aplikaci na této technologii je také nutná alespoň elementární znalost technologie Docker a NodeJS nebo Golang programovacího jazyka pro definování parametrů pro následné zpracování datových bloků uvnitř blockchainu.

Z hlediska použití Hyperledger Fabric je velmi jednoduchá plug-and-play technologie, která jde v případě rozsáhlejších znalostí i jiných technologií škálovat na velmi komplexní a specifickou aplikaci. Tato technologie proto bude i následně zvolena v praktické části jako ta, na níž budeme řešit náš problém, více informací viz [2, 6, 7].

Quorum

Quorum je další možná technologie, na kterou může být aplikován blockchain. Je založena na technologii Etherea a nabízí možnou aplikaci jak privátního, tak veřejného blockchainu. Ve veřejné verzi ovšem nevyužívá žádné důkazy v rámci konsensuálních algoritmů pro ověření správnosti dat.

Toto je také důvod, proč je samotná aplikace nějakého problému na této technologii velmi složitá a je nutno přinést mnoho bezpečnostních vylepšení, aby mohla být tato technologie aplikována na daný problém. Quorum tedy nebudeme zvažovat pro praktickou část, více informací viz [6, 7].

3.3 Limity blockchainu

Blockchain jako každá nová technologie má svá pro i proti. Zatím jsem se v práci spíše zaměřoval na ty vlastnosti, jež nám poskytuje nad rámec starých technologií, v této kapitole se proto zaměřím na limity a rizika, které je potřeba zvážit během implementace blockchain technologie, více informací viz [13].

Ověření

Mezi hlavní limity blockchain technologie patří nutnost ověření podpisem, redundance, energetická náročnost, bezpečnost a zákony. Jelikož blockchain slouží, jako decentralizovaná databáze, je nutné určitým způsobem zajistit ověření jednotlivých účastníků do databáze. K tomu slouží již zmíněná asymetrická kryptografie, která je komplexní a velmi časově náročná v porovnání s klasickým ověřením uživatel a heslo. Společně s ověřením podpisem je v blockchainu nutné, aby všichni účastníci dodržovali předem nadefinovaný konsensuální algoritmus a mohli tak pomáhat s ověřováním a ručením za správnost dat v blockchainu, více informací viz [13, 14].

Redundance

Redundance je další problém, jenž se týká výkonnosti blockchain databáze. Pro každý prvek, respektive záznam do blockchainu musí být provedeny potřebné bezpečnostní operace pro všechny již vzniklé záznamy do blockchainu. Toto může zvýšit

časovou náročnost a správu. U centralizovaných databází dochází vždy jen ke správě aktuálního prvku, a nikoliv celé databáze, více informací viz [13, 14].

Energetická náročnost technologie

Energetická náročnost pro fungování blockchain databáze je velmi vysoká, protože každý připojený účastník musí mít vlastní kopii dat a musí ji neustále ověřovat přes výpočty skrze asymetrickou kryptografii. V případě Bitcoinu se už ani nevyplatí těžit ho výpočetním výkonem, protože množství přidělených bitcoinů a spotřebovaná energie už nejsou výdělečné, více informací viz [13, 14].

Bezpečnost

Blockchain technologie bude vždy náchylná k tzv. 51% útoku. Tato technologie je založena na kooperaci a důvěryhodnosti jednotlivých účastníků v databázi. Proto při menším počtu účastníku roste riziko možného podvržení dat v databázi, a to právě získáním více jak poloviny hlasů a následným schvalováním falzifikovatelných záznamů v databázi. Bezpečnost technologie blockchain je také velmi úzce svázaná s možnou lidskou chybou, kdy při vstupu nejsou data nijak předzpracovaná. Součástí diplomové práce bude tedy implementace ošetření všech chybových stavů, které mohou nastat, více informací viz [13, 14].

Zákon

Jedna z největších výhod blockchainu je možnost decentralizované kontroly dat. Tato data musíme ale vždy nějakým způsobem adresovat, případně účastníci musí prokázat vlastnictví. Tato nutnost je v rozporu s obecným Nařízením Evropského parlamentu a Rady č. 2016/679 o ochraně fyzických osob žijících v Evropě. Pod zkratkou GDPR platí v Evropě platí od 25.5. 2018.

Hlavním úkolem GDPR je právě snaha o regulaci kontroly osobních dat, která sdílíme na internetu např. v případě nákupu v elektronickém obchodě. Motivem pro vytvoření této ochrany byl také skandál firmy Cambridge Analytica, která shromažďovala a obchodovala s daty, která získala bez souhlasu. Osobní data jsou všechna data, která mohou identifikovat osobu, ať už je to přímo, nebo nepřímo, a vedou tedy k identifikaci osoby. Nemusí to být tedy pouze jen jméno, které přímo odkazuje na osobu, ale také ID číslo, které je 1:1 přiřazené k osobě. Článek 17 pak také definuje, že by entita, která shromažďuje citlivá data, měla být schopná odstranit tato data bez jakéhokoliv prodlení, pokud o to uživatel požádá. Toto pravidla limituje mnoho možných aplikací ve veřejném blockchainu. V rámci privátního se dá do smlouvy před vstupem do sítě blockchain dát ujednání o tom, jaká data budou

uložena a také jak s nimi bude nakládáno a jak budou zabezpečená v případě pokusu o získání těchto dat skrze třetí osobu. Velká překážka pro rozšíření blockchain implementací je i to, že data vytvořená skrze asymetrickou kryptografii a sdílený veřejný klíč budou i tak považovány za dostačující k následné identifikaci osob. Do budoucna se tedy uvažuje o zavádění rušení do dat nebo výplňových dat, které znemožní identifikaci osob. Anonymní kryptoměny jako Monero nebo Zcash už fungují na kompletně anonymní implementaci, ale odstranění dat je problém. Budoucnost v blockchainu určitě pro nějaké implementace bude, jen je nutné, aby právní orgány komunikovali s technologickými experty, aby nebyl technologický pokrok zpomalován komplikovanou legislativou, více informací viz [13, 15].

4 Bezpečnost dat

Šifrování jako součást kryptografie slouží k zabezpečení originálních zpráv, aby k nim třetí osoba nemohla získat přístup. Správa přístupu k datům pak následně může být rozdělena na ochranu před změnou dat, ochranu před čtením dat, ověření o původnosti dat a nepopiratelnosti autenticity zprávy - toto řeší zabezpečení pomocí digitálního podpisu. Samotné šifrování v rámci kryptografie tedy slouží k zašifrování čistého textu (plain text) a vytvoření zašifrovaného textu (cipher text). Takto provedeným šifrováním, ať už symetrickým, či asymetrickým, může text následně dešifrovat pouze ten, kdo zná klíč, nebo pokud byl nasdílený veřejný klíč, s jehož pomocí byl odvozen privátní klíč, který zprávu zašifroval.

Během šifrování dojde k transformaci dat za účelem zabezpečení. Součástí šifrování je hashování v případě využití i HMAC kódu. Hashovací funkce je matematická funkce, která z různě dlouhých vstupních dat vždy vytvoří výstupní data pevné délky, přičemž stejné vstupy mají vždy stejné výstupy. Při jakékoliv změně vstupu dojde ke změně výstupu. Hashování se využívá pro zvýšení integrity dat, kdy máme v datech obsažené kontrolní součty. Příkladem hashovacích funkcí jsou například MD5, SHA, SHA-128/256.

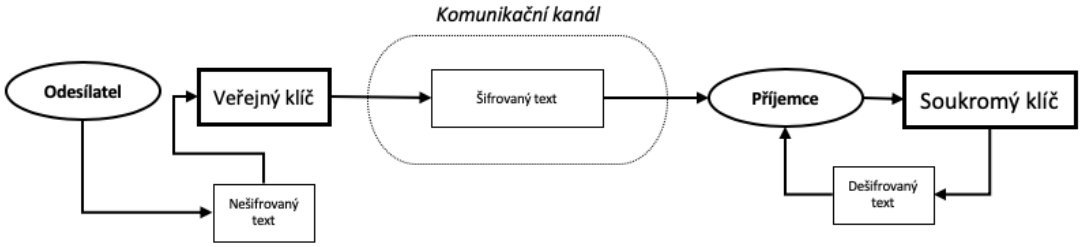
Za celým procesem kryptografického algoritmu nebo také šifry je několik přesně definovaných kroků. Při šifrování je stěžejní využití klíče. Klíč může být sdílený a tajný, pak jde o symetrické šifrování, při implementaci asymetrické kryptografie jsou využity 2 klíče, a to veřejný klíč a soukromý klíč. Tyto 2 jsou od sebe vzájemně odvozeny. Síla šifry pak také závisí na počtu bitů, se kterými jednotlivý algoritmus pracuje, více informací viz [16, 17].

4.1 Symetrické algoritmy

Symetrické algoritmy využívají pro šifrování a dešifrování stejný klíč. Pokud tedy chceme přenést šifrovanou zprávu druhé straně, musí být bezpečně přenesen i sdílený tajný klíč k následnému dešifrování. Symetrické šifry pak můžeme ještě dělit na proudové, kde se pracuje s nepřetržitým proudem symbolů, například algoritmus RC4. Druhý způsob představují blokové symetrické šifry, kde se pracuje s bloky bitů. Jde například o DES a AES. Tyto algoritmy jsou bezpečnější vůči útoku. Pro funkčnost symetrické kryptografie je nutný přenos klíčů. Nutný naopak není u asymetrické kryptografie, a proto je i více používána, více informací viz [16, 17].

4.2 Asymetrické algoritmy

Asymetrická kryptografie používá pro šifrování dva různé klíče, aby eliminovala riziko při přenosu klíčů v symetrických šifrách. Klíč, kterým zprávu šifrujeme, je veřejně známý, ale klíč na dešifrování je privátní. Mezi privátním a veřejným klíčem jsou matematické vazby, s jejichž pomocí vznikne jejich odvození při vytváření a na základě prvočíselných vlastností čísel je možné zprávy šifrovat a dešifrovat. Tento způsob šifrování a dešifrování je velmi výpočetně náročný a využívá se například k distribuci klíčů pro symetrické šifry. Tato kombinace obou způsobů se využívá například u TLS zabezpečení. Na obrázku níže vidíte princip asymetrického šifrování, více informací viz [16, 17].



Obr. 4.1: Asymetrické šifrování a dešifrování

4.3 Použití a doporučené algoritmy pro šifrování

Úrovně bezpečnosti šifry se měří na délku doby prolomení a získání klíče. Přesněji to pak udává hodnota 2^n, kdy n je počet bitů klíče. Algoritmus SHA-128 má například hodnotu 128. Nejkratší klíče jsou pak naopak zastoupeny právě v symetrických algoritmech, kde například u DES je to délka 56 bitů a u asymetrického algoritmu RSA až 3072. Doporučení organizace NIST pro rok 2020 je 2048 bitů označeno za bezpečné vzhledem k výpočetním výkonům dnešních počítačů, více informací viz [16, 17].

5 Konsenzuální algoritmy

Při vytváření blockchain sítě a databáze musíme nějakým způsobem odlišit velké množství účastníků a jejich transakce a navíc rozhodovat o správnosti každé z nich. K tomu slouží právě konsenzuální algoritmy. Podstatou je, že máme skupinu připojených účastníků, kteří si navzájem ověřují správnost obsahu v databázi a dostávají za to nějakou odměnu, více informací viz [2, 8].

5.1 Konsenzuální model Blockchain

Aby takto vytvořený model mohl fungovat, musíme vytvořit základní pravidla a povinnosti každého účastníka v databázi. Plnění povinností se v rámci různých technologií liší, ale pravidla a povinnosti z bodů vypsaných níže.

Rozhodování

Mechanismus konsenzuálního algoritmu musí sbírat všechna užitečná data, které mohou změnit obsah v databázi.

Kolaborace

Každý účastník musí být připojen do sítě za účelem, že jeho účast bude pro všechny výhodná a pomůže se správným rozhodováním nad transakcemi.

Kooperace

Každý jednotlivý účastník musí své vlastní zájmy vzhledem k výhodnosti transakcí a pozici v síti dát stranou a soustředit se na zájmy všech účastníku síti a jejich prospěch.

Rovnost účastníků

Všichni účastníci mají stejná práva i hlasovací hlas, to zajistí, aby byly všechny hlasy účastníků v síti stejně důležité a aby pro každého mělo smysl v síti setrvat jako aktivní účastník.

Povinná účast

Každý účastník musí být v síti přítomen a musí se účastnit hodnocení všech transakcí. Toto zaručí, že nikdo nezůstane mimo hlasování, takže bude zaručena 100% správnost a ověření transakcí.

Aktivita

Účastník je stejně aktivní jako každý jiný. Není zde možnost být více aktivní a mít tím pádem větší odpovědnost ve skupině. Vyjma jeden speciální konsensuální algoritmus, který bude zmíněn dále v další kapitole, více informací viz [2, 8].

5.2 Typy konsenzuálních algoritmů

Konsenzuální algoritmy jsou tedy algoritmy, jež nám zajistí spolehlivost a ověření dat v síti, kde se mohou vyskytovat pokusy o podvržení dat, která jsou sdílena mezi účastníky. Výhod využití právě konsenzuálních algoritmů je několik. Jde o možnost real-time ověřování dat. V případě chyby nebo podvržení dat je velmi jednoduché data opět zpětně nastavit a odstranit chybu a další.

Existuje velké množství různých algoritmů, které se mohou hodit vždy na jinou aplikaci blockchainu. Shrneme tady ty hlavní a nejčastěji využívané typy konsenzuálních algoritmů, více informací viz [8].

Proof-of-Work

Jde o nejpopulárnější algoritmus využívaný například kryptoměnami, jako je bitcoin. Účastníci tohoto konsenzuálního algoritmu si říkají těžaři a mají za úkol ověřit a potvrdit transakce v databázi. S každou transakcí, kterou ověří výpočtem nového bloku, získávají těžaři odměnu - v případě bitcoinu jde o malou část této měny.

Odměny jsou udělovány na základě toho, že do sítě věnují svůj výpočetní výkon a elektřinu pro vyřešení problému. Tímto způsobem musí měna také fungovat, aby byly neustále ověřovány transakce probíhající v síti.

Těžaři využívají hashovací funkce pro nalezení dalšího místa pro vložení transakce do sítě. Jakmile najdou výsledek, transakci vloží dovnitř a následně jsou odměněni. Čím více dat do samotné sítě vložíme, tím je stále těžší vypočítat následující hash hodnoty. Eventuálně tímto způsobem například u bitcoinu dojdeme na konec rozsahu a tato měna bude jednoho dne zcela vytěžena. Aktuální předpoklady počítají s vytěžením až v roce 2140, více informací viz [8, 10, 9].

Proof-of-Stake

Jde o jiný přístup k ověřování transakci v blockchainu. Hlavní rozdíl od Proof-of-Work spočívá v tom, že ověřování transakcí probíhá způsobem dokazování vlastnictví jednotlivého bloku dat na nově vytvořeném bloku. Na příkladu kryptoměn jde

o způsob vykázání množství měny, kterou na svém účtu držíme, a následné porovnání v rámci sítě pak už jen určí, zda se celková suma všech účastníků shoduje. Poté se aktuální stav sítě prohlásí za správný, více informací viz [8, 10, 9].

Proof-of-Importance

První platforma, která implementovala Proof-of-Importance, byla NEM. Díky tomuto přístupu kombinujeme oba předchozí algoritmy do jednoho, jak dokázání vlastnictví, tak i hodnotu hashe pro další transakci a následnou odměnu. Každý účastník dostane mnohem větší práva a následně důvod, proč se aktivně účastnit provozu v síti. Ke všem účastníkům se přidává položka skóre a ta hodnotí, jak moc je každý aktivní, a zvyšuje jejich důležitost v síti. Jako bonus k tomuto konsensuálnímu algoritmu je také navíc rychlejší přístup k vytváření nových bloků pro účastníky právě s vyšším skóre, více informací viz [8, 10, 9].

Proof-of-Activity

Výše zmíněné konsenzuální algoritmy jsou ty nejčastěji používané pro aplikace na jednotlivé příklady v už uplatněné praxi. Existuje ovšem mnohem více alternativ. Proof-of-activity kombinuje Proof-of-Work a Proof-of-Stake. Každý z předchozích má tedy minimálně 51 % účastníků správně interpretující informace jako hladinu na podvržení dat a aby byli tedy následně označené za správné. Díky volbě tohoto algoritmu ji tedy ještě zvedneme. Nezavádí se zde ale pole pro skóre, jako Proof-of-Importance, více informací viz [8, 10, 9].

Proof-of-Burn

Další a velmi zajímavý konsenzuální algoritmus je Proof-of-Burn. Potvrzení o správnosti pouze jednoho výskytu informací o transakcích je prováděno tzv. pálením nebo zničením bloku. Tento algoritmus se snaží eliminovat energetickou náročnost Proof-of-work, ale přitom zachovat srovnatelnou bezpečnost. Smazáním klíče pro přístup k určitému množství měny dostaneme tedy potvrzení o tom, že se opravdu taková hodnota na uživatelském účtu nachází. Jakmile je smazána, stačí pak ji už jen připsat na jiný účet, více informací viz [9, 10].

Proof-of-capacity

Tento algoritmus opět vychází z velmi populárního algoritmu Proof-of-Work. Místo výpočetní síly ale využívá volné místo na disku. Energetická náročnost pro funkčnost provozu tohoto typu blockchainu bude tedy mnohem menší. To je tedy mnohem šetrnější k přírodě, jelikož v rámci 21. století je každý dopad nové technologie ve

výpočetní technice na životní prostředí velmi diskutované téma. Pro zapsání dat do blockchainu musí být vždy alokováno určité místo na disku, které poskytne této síti prostor na odložení nějakých dat, a následně je označený blok potvrzen za správný. Velikost alokovaného prostoru se opět mění a informace o ní jsou zasílány opět asymetrickou kryptografií, více informací viz. [8, 10, 9].

Konsensuální algoritmus	**Technologie**	**Implementace**
Proof-of-work	Záznamy všech transakcí	Bitcoin
Proof-of-stake	Prokazování vlastnictví vlastněné měny	Ethereum
Proof-of-importance	Kombinace proof-of-work a proof-of-stake, skóre pro úspěšně ověřené transakce	NEM
Proof-of-activity	Kombinace proof-of-work a proof-of-stake	Espers
Proof-of-burn	Eliminace starých bloků	Slimcoin
Proof-of-capacity	Alokování prostoru na disku	Filecon

Tab. 5.1: Přehled konsenzuálních algoritmů

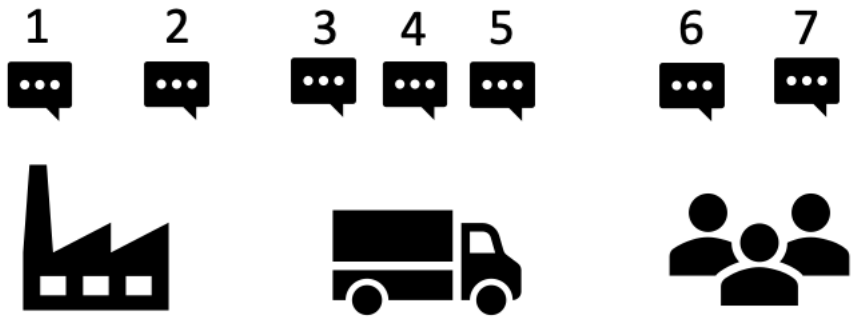
5.3 Rizika

- Jakýkoliv distribuovaný systém je náchylný k chybám a mohou zde nastávat i rizikové stavy jako například pokusy o podvržení dat útokem pomocí multiplikace účastí v blockchain síti. Správně definovaná síť bude schopna některé tyto stavy eliminovat, a nebo jim i předejít správnou implementací samoopravných metod. Během komunikace může dojít ke ztrátě informace a zpráva musí být následně znovu poslána.
- Byzantská chyba neboli chyba, kdy se jednotliví účastníci chovají jinak, než bylo očekáváno. Jde o členy, kteří nekomunikují dle pravidel pro správné fungování decentralizovaného systému. Takto vytvořená skupina chce pak podvrhnout data vytvořená ostatními účastníky blockchainu. Tato chyba není možná v kontrolovaných prostředích, jako je ten od SAP, Google nebo třeba od Amazonu, více informací viz [10].

6 Implementované blockchain aplikace

Blockchain není technologie jen pro vývoj kryptoměn, ale také nabízí implementační řešení pro problémy za účelem urychlení logistických procesů a mnoho dalších. Mezi nejznámější implementace blockchainu patří tzv. spotřebitelský řetězec. Tato implementace ukládá během cesty potravin informace do blockchainu, kde se recipient může následně například přesvědčit o udávaném místě původu zboží. Tato implementace může sloužit velmi dobře pro potravinářský, ale i farmaceutický průmysl, kde je velmi nežádoucí prodávat jakékoliv falzifikáty. Toto není jediný příklad implantace v lékařství. Při ukládání více osobních dat se zvýšeným způsobem zabezpečení by mohly být například zdravotní karty ukládány do blockchainu, ke kterému by v případu nutnosti mohla přistoupit například jakákoliv nemocnice. Stačí jen vyřízení přístupu a oprávnění k přístupu do blockchainu, který bude mít uložené všechny informace k dispozici pro jakéhokoliv doktora. Aby tato implementace mohla být využívána, je žádoucí, aby měla velmi bezpečné spojení pro ověření identity.

Velmi zajímavá implementace technologie je například také fintech startup firma Helperbit, která se zaměřuje na zvýšení transparentnosti v rámci humanitárních příspěvků. Každý přispěvatel pak má možnost kdykoliv se podívat, kdy a kam byly jeho peníze poslány. Tato implementace funguje na technologii Bitcoin a popularita zvýšení transparentnosti v tomto specifickém příkladě může vést i k vyšší získané finanční pomoci, a to zvlášť kvůli vyšší atraktivnosti díky transparentnosti technologie blockchain, více informací viz [19, 22].



Obr. 6.1: Časová razítka

Na obrázku výše vidíme implementaci právě pro spotřebitelský řetězec, kde je vždy při nějakém vstupu do nového stavu na čárový kód zaznamenáno časové razítko, lokace, teplota nebo jiné vybrané parametry pro zvýšenou kontrolu kvality vývozce. Na příkladu zde jde tedy o časové měřítko celkem na 7 místech jednotlivé objednávky.

V každém bodě je zaznamenán stav a data jsou odeslaná do blockchainu. Zpětně si tedy každý účastník připojený do této sítě může zjistit v, jakém stavu a kde přibližně se zboží nachází. Tato transparentnost je v případě distribuce zboží velmi hodnotný údaj, je tedy rozšířen i mezi velké firmy na světě, příkladem může být jakékoliv zboží, které je chráněno označením původu, více informací viz [19].

6.1 Způsoby integrace blockchain technologie

Implementace blockchainu není tedy pouze vytváření nějaké měny a následně odměňování těžařů a následné spekulování s měnou. Samotná implementace technologie blockchain může být založena i na využití chytrých kontraktů nebo také na možné integraci zařízení z internetu věcí, více informací viz [19].

Integrace transakcí

Využití blockchainu jako způsobu placení a zaznamenávání počtu měny na účtech má velké výhody, jichž je docíleno právě decentralizací. Na technologii blockchain můžeme vytvořit vlastní kryptoměnu a využívat ji, jak budeme chtít - bez nutnosti správy centrální entitou. Samotný Bitcoin je open source platforma a nabízí jednoduché API pro implementaci a pro spuštění platformy k placení touto měnou pro jakéhokoliv konečného zákazníka, více informací viz [19].

Chytré kontrakty

Chytrý kontrakt sám o sobě je jen samořídicí automat, který rozhoduje mezi pravdou a lží. Tímto způsobem můžeme právě na Ethereu vytvořit implementace blockchainu, které umožňují práci s daty v blockchainu a následně řetězení reakcí v případě přechodu do nějakého stavu. Tímto způsobem se dá eliminovat třetí strana, která nemá například na obchodu kromě potvrzení žádný jiný zájem. Implementace chytrých kontaktů je díky pokročilému vývoji platformy Ethereum velmi jednoduchá. Mezi tyto implementace se řadí i toto řešení, následně uvedené v praktické části této diplomové práce. Mezi zajímavé nápady implementace patří také možnost využití blockchain sítě při volbách. Každý člověk může vykonat pouze jeden hlas vzhledem k tomu, že vkládá své jedinečné ID a následně volí. Zamezilo by se tedy chybným hlasům a ulehčilo by administrativní zátěž zvlášť při sčítání hlasů. Navíc každý zvolený hlas by nebylo možné nikdy změnit díky tomu, že technologie blockchain neumožňuje změnu dat bloků nebo čistě jejich odstranění, více informací viz [19].

Zařízení internetu věcí

Implementace blockchain technologie pro zařízení patřící do internetu věcí by mohla přístupem do různých sítí reagovat včas na definované podněty. Propojením jednotlivých zařízení v domácnosti skrz blockchain by mohlo několik zařízení komunikovat spolu a následně by mohli vykonat nějakou společnou sekvenční aktivitu. Tato automatizace by mohla docílit nezávislé kooperace zařízení v domácnosti bez lidské intervence. Připojení velkého množství senzorů, zařízení a jejich následné doplnění informací v procesu spotřebitelského řetězce by opět zvýšilo transparentnost celého procesu. Mnoho implementací pro tuto technologii nevzniklo, ale vidím v ní možnou budoucnost, více informací viz [19].

Ochrana ID na základě blockchainu

Ukládáním informací pro přihlášení do systému skrze blockchain strukturu bychom získali nezávislou ověřovací entitu, z níž nelze odcizit žádná data. Vždy by šlo o ověřování dotazem na správné přihlašovací jméno a heslo. Správná kombinace by pak umožnila připojení do sítě. Tato implementace je také podporována několika společnostmi, například startup Validate ID vytvořila platformu pro správu ID přístupových informací, více informací viz [19, 26].

6.2 Kryptoměny

Jako jeden z prvních příkladů lze označit využití decentralizované databáze a blockchain technologie v praxi pro kryptoměny. Kryptoměna je typ digitální měny, která není kontrolována žádnou centrální entitou. Tyto měny byly tedy vytvořeny za účelem zvýšení transparentnosti finančního systému, zvýšení rychlosti a bezpečnosti transakcí. Základ všech kryptoměn je řetězení digitálních podpisů a transakcí, kde se využívá výpočet kryptografických funkcí, za něž jsou těžaři odměňováni stanovenou finanční odměnou – blockchain. Kryptoměny nejsou kryty zlatem jako klasické měny, ale jsou vytvořeny tak, aby padělání nebo ovlivnění hodnot předchozích transakcí bylo teoreticky nemožné, a jsou zcela nezávislé na klasických měnách. Na světě nyní existuje přes 1000 různých kryptoměn a tržní kapitalizace nejrozšířenější z nich – bitcoinu přesahuje hodnotu 150 miliard dolarů. Rozdíly mezi různými typy nejrozšířenějších kryptoměn budou popsány dále v diplomové práci, více informací viz [19, 20, 21].

Bitcoin

Bitcoin je nejrozšířenější kryptoměna, která slouží především k placení. Vznikl v roce 2008 a v oběhu je pouze 21 milionů kusů. Od začátku fungování se technologie nezměnila. Bitcoin funguje na konsenzuálním algoritmu proof-of-work. Pro ověření hodnoty následujícího bloku se pro těžaře využívá algoritmus SHA-256. Pro těžení a následné ověřování transakcí v Bitcoinu síti je zapotřebí výkonu ASCIs. Bitcoin postrádá možnost škálovatelnosti v porovnání s konkurenčními kryptoměnami, ale jeho možné využití může být k uložení financí díky omezenému množství mincí a dostačující bezpečnosti, více informací viz [19, 20].

Ethereum

Ethereum není kryptoměna založená pouze na platebním systému, jako předchozí bitcoin. Na rozdíl od něj přináší chytré kontrakty a umožňuje více možností využití blockchain technologie. Ethereum funguje na konsenzuálním algoritmu Proof-of-work, ale do budoucna se jedná o přechod na Proof-of-stake. Některé implementace už tento algoritmus využívají. Proof-of-stake v kombinaci s Casperovým algoritmem, který donutí všechny účastníky potvrzovat data v blockchainu, eliminuje byzantskou chybu, která je u předchozího konsenzuálního algoritmu kritická. Etherum nemá omezený počet mincí a zároveň velikost jednoho bloku je tak malá, že se nepočítá, že bude někdy dosaženo limitu pro maximální počet transakcí. Proto je Ethereum vhodné pro rozsáhlé implementace, více informací viz [19, 20].

ZCash

ZCash byla jedna z prvních kryptoměn, která vychází z technologickém vzoru Bitcoinu. Hlavním cílem pro ZCash je poskytnutí absolutní anonymity lidem platícím s touto kryptoměnou. Konsenzuální algoritmus i konečný počet mincí je zde stejný jako v Bitcoinu, ale je tady navíc rozšířen o kryptografický protokol zk-SNARK, který zajišťuje absolutní anonymitu odesílatele i příjemce této měny, více informací viz [18].

Ripple

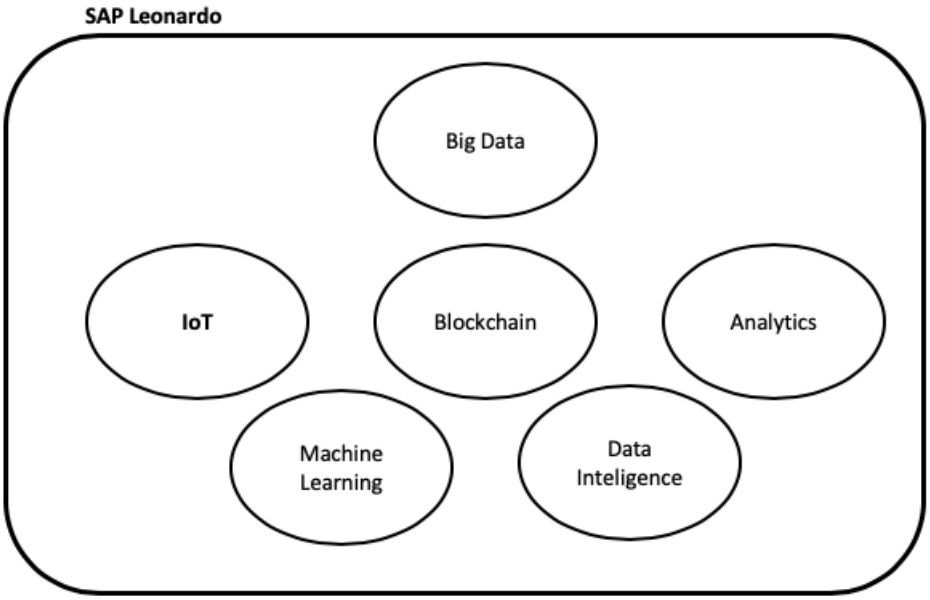
XRP neboli Ripple je kryptoměna, která byla vytvořena společností Ripple za účelem vyšší rychlosti a větší bezpečnosti všech transakcí než má Bitcoin. Neslouží k přenosu pouze peněz, ale také pro přenos digitálních dat. Fungování je podobné jako SWIFT pro bankovní komunikaci. Ripple je využívaný spíše bankami než jednotlivci připojenými do sítě. Konsenzuální algoritmus očekává od ostatních účastníků pouze potvrzení transakce skrze chytré kontrakty, více informací viz [19].

Litecoin

Litecoin je další kryptoměna velmi podobná bitcoinu. K dispozici má ovšem mnohem větší množství mincí, a to 84 milionů. Litecoin využívá stejný konsenzuální algoritmus, ovšem pro ověřování následujících bloků nepoužívá algoritmus SHA-256, ale Scrypt. Pro ověřování transakcí pak následně není třeba žádného speciálního obvodu, ale stačí pouhá výkonnostní síla CPU, případně GPU. To by mělo tuto kryptoměnu rychleji rozšířit mezi veřejnost a zajistit tak ještě rychlejší potvrzování transakcí. Průměrná doba pro potvrzení transakce je 2,5 minuty, což je 4krát rychlejší než bitcoin, a to díky jednoduchosti těžení pro každého uživatele, více informací viz [19, 21].

7 Portfolio SAP

SAP Leonardo je skupina produktů v rámci nových technologií produktového portfólia SAP. Jde o technologie, které mohou mít zajímavé a často revoluční řešení pro zastaralý proces nebo mohou zákazníkům nabídnout nové informace, kteří ti je potom mohou využít. Leonardo nabízí služby pro strojové učení, inteligentní správu dat, analytické nástroje, internet věcí nebo také blockchain. Tyto služby a získaná data z nich se dají i následně propojovat navzájem, a to například pomocí konektorů na HANA databázi. Toto následně přinese výhody a kooperaci jednotlivých služeb, které zvyšují funkčnost implementovaných technologií na portfoliu od SAP, více informací viz [11, 12].



Obr. 7.1: Portfolio SAP Leonardo

7.1 SAP Cloud Platform

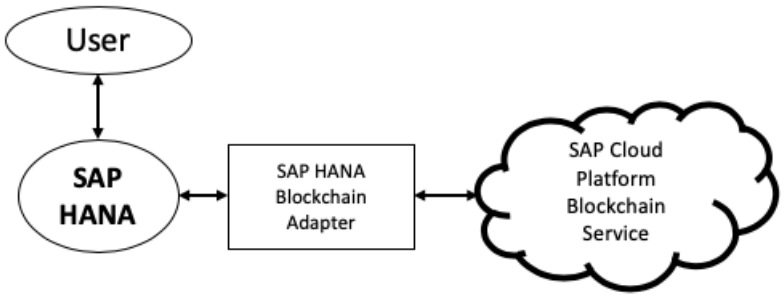
SAP Cloud Platform je platforma, která umožňuje implementaci námi zvolené technologie na modelu Software as a Service (SaaS). Tento model funguje na principu předplatného za jednotlivé služby, případně pouze za instance služeb, které se aktuálně používají. Platí se tedy pouze za to, co používáme, nikoliv za další moduly, které k funkčnosti naší aplikace v rámci SAP Leonardo nevyužíváme. SAP Cloud Platform je využíván řadou velkých firem, jako jsou například Cisco nebo Bosch, více informací viz [11].

7.2 SAP Cloud Platform Blockchain Service

SAP Cloud Platform Blockchain Service je služba poskytovaná jako SaaS. Nabízí nám velmi lehký a bezpečný způsob, jak si testovat a následně uvést do provozu námi vytvořenou decentralizovanou databázi v cloudu. Blockchain services získáme předplacením námi využívaného počtu zařízení připojených do databáze a také podle počtu a paměti využívaných samotnými účastníky. Cena za tuto službu je vyšší a to z důvodů kompletní možné integrace na jiné technologie poskytované od SAP a také poskytnutí zrychlení přenosu informací při zajištění 100% ochrany právě díky výběru SAP Cloud Platform pro implementaci. Pro vyzkoušení služeb máme k dispozici také trial verze, a to přímo na webu Cloud Platform, takže si může každý vyzkoušet nadefinovat svůj vlastní blockchain a také pravidla ochrany s jedním připojeným účastníkem. Toto seznámení postačí na vytvoření zkušebního modelu právě pro úsek praktické části této diplomové práce, více informací viz [11, 12].

SAP HANA Blockchain Service

SAP HANA Blockchain Service je další služba fungující v kooperaci s blockchainem, vytvořená na Cloud Platform. Použitím SAP HANA Blockchain Adaptér můžeme propojit data zadaná do SAP HANA databáze a následně je sdílet a přidat i možnost pro úpravu různým účastníkům napojeným do vytvořeného blockchainu.



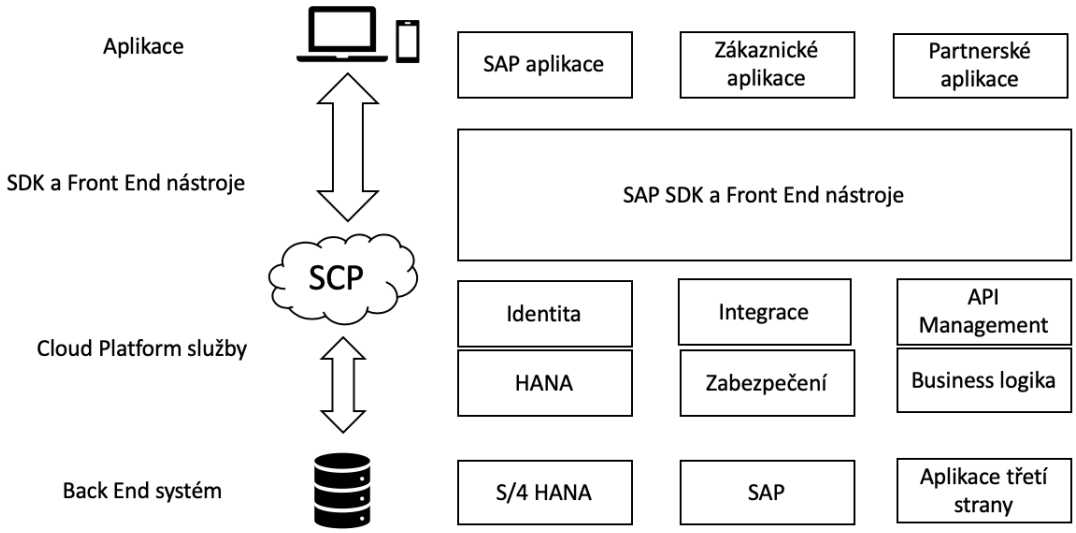
Obr. 7.2: Propojení s HANA databází

Propojování a správa dat v SAP Leonardu jako celku jde velmi jednoduše, a to pomocí veřejně přístupných API metod, které se pak volají ve spustitelném programu. Ten pak běží už nad samotnou databází v cloudu.Takže následná implementace umělé inteligence nebo jiných modulů pro kontrolu nebo sběr dat z blockchain databáze jsou také velmi jednoduché a otevřené pro uživatele, více informací viz [11, 12].

Sada pro vývoj softwaru - SDK

Jedna z dalších možných integrací již existujících aplikací ze SAP Cloud Platform do mobilních operačních systémů je právě díky dostupné sadě pro vývoj softwaru, zkráceně SDK. Tato sada nám umožňuje skrze jednoduché volání API přistupovat ke všem modulům naší aplikace a na druhou stranu i k ověřovacím metodám systému iOS, jako je například touchID. Všechna tato integrace je vytvořena už v programovacím jazyce Swift, který slouží k vývoji aplikací pro iOS. Toto může být malá překážka, protože pro vývoj na mobilní systém android se dál pokračuje v Javě a byla by nutná kódová konverze.

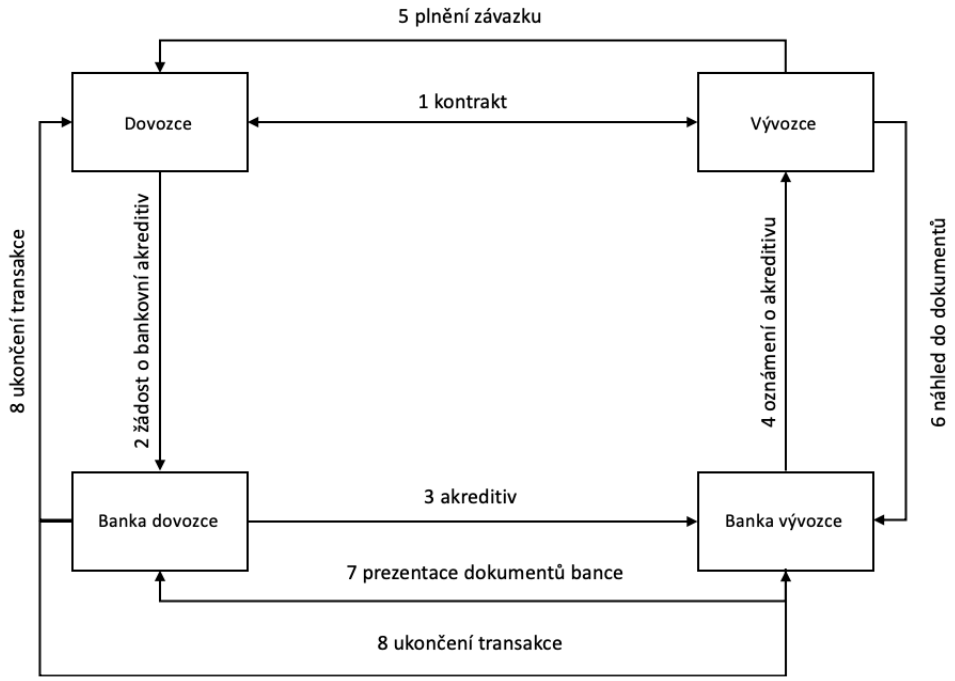
Na obrázku níže je zobrazen celý vývojový proces a možné implementační vylepšení, které můžeme přidat pro naši aplikaci na SAP Cloud Platform SDK, více informací viz [12].



Obr. 7.3: Schéma vývoje aplikací na SCP

8 Bankovní akreditiv

Bankovní akreditiv se využívá jako nástroj pro platbu a zajištění převedení financí od příkazce až k beneficientovi. Beneficient přitom vždy obdrží dohodnutou výši odměny včas, i kdyby příkazce neměl dostatek financí, protože je to krytá částka díky smlouvě s bankou příkazce. Bankovní akreditiv se nejčastěji využívá v rámci mezinárodního obchodu, kde rozdílná legislativa a neznalost obchodních stran tyto transakce prodlužuje. Poskytnutí této služby je samozřejmě placené odpovídajícími procenty z částky, která je takhle pojištěna každou bankou, a možný přechod správy přes celý proces do blockchainu by tyto poplatky snížil a urychlil celý proces, více informací viz [23, 25].



Obr. 8.1: Bankovní akreditiv

8.1 Princip akreditivní transakce

Celý proces pro bankovní akreditiv je vidět na obrázku výše. Skládá se z 8 kroků, které se musí dokončit před tím, než může být obchod prohlášen za dokončený. První krok je, když obě strany mezi sebou uzavřou smlouvu o obchodu, kde jsou definovány

všechny nutné součásti, jako jsou cena, termíny plnění, dodání a jiné. Poté následuje druhý krok, kdy dovozce informuje svoji banku o žádosti o vytvoření bankovního akreditivu a uzavřou spolu smlouvu. Banka dovozce následně vytvoří smlouvu a připraví dokumenty pro založení bankovního akreditivu pro banku vývozce. To už je součástí druhého kroku. Třetí krok je už následná prezentace podkladů pro banku vývozce pro zajištění plnění závazku. Ve čtvrtém kroku banka vývozce informuje vývozce o vytvoření bankovního akreditivu. Ten pak slouží jako záruka o splacení obchodu, a proto vývozce postupuje k pátému kroku, kdy jde už o odeslání zboží nebo vykonání služby, která je předmětem obchodu. Následně pro získání celé odměny za vykonání závazku prezentuje vývozce podklady své bance o provedení závazku. V sedmém kroku pak banka vývozce tyto dokumenty jen přepošle bance dovozce. V osmém kroku jsou tato data následně prezentována dovozci a všechny finanční závazky jsou naplněny a obchod může být prohlášen za dokončený, více informací viz [23, 24, 25].

8.2 Nevýhody akreditivu

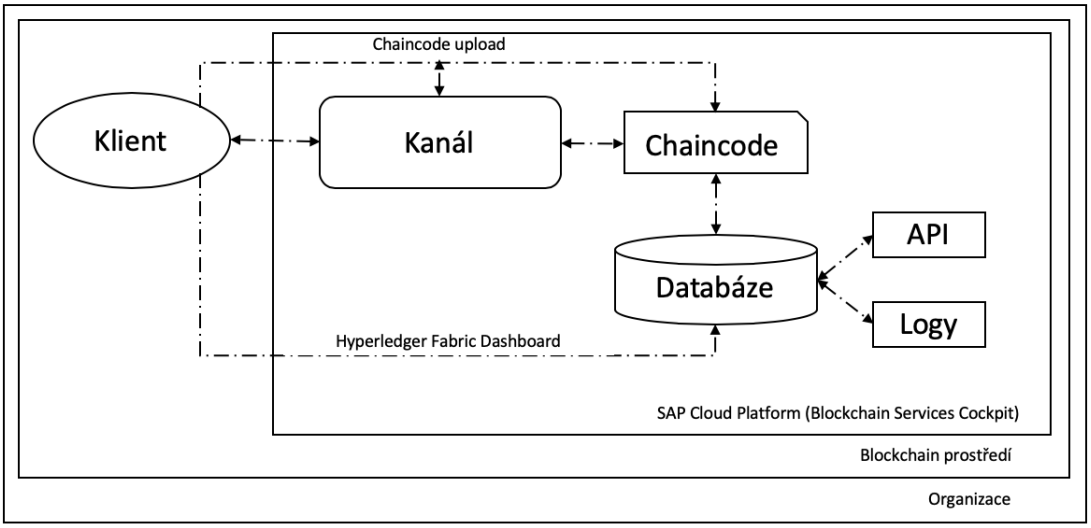
Jde o bezpečný, ale vcelku komplikovaný způsob, jak se dá zajistit vysoká bezpečnost a plnění závazků vzájemně mezi oběma obchodními subjekty. Bankovní akreditiv ale nemá jen výhody. Mezi nevýhody například patří, že se velmi často dodržují pevně dané termíny pro vykonání jednolitých kroků a následné opoždění může způsobit vysoké pokuty kvůli nedodržení termínu, i když jsou zaviněné třeba i třetí stranou. Další nevýhodou jsou poplatky bance, již se odvíjí od výše transakce, kterou banka kryje svými vlastními zdroji. Proto je velmi důležité zvážit, zda nám tento způsob obchodu vyhovuje a zda se vyplatí mít peníze kryté. Přesunutím správy nad tímto obchodem do blockchainu by měly všechny strany ihned informace o tom, v jaké fázi je obchod a jaký krok následuje. V rámci diplomové práce půjde tedy o velké ulehčení správy tohoto procesu za využití moderní technologie, více informací viz [23, 25].

9 Implementační část

V rámci praktické části bude vytvořena implementace řešení na základě nápadu na inovaci ve firmě SAP. Funkční implementace bude podporovat správu celého procesu pro bankovní akreditiv. Všechny strany účastnící se tohoto procesu budou mít možnost náhledu na celý proces. Všechny součásti této komunikace budou vždy uloženy a ověřeny, zda jsou validní pro zajištění bezpečnosti komunikace.

9.1 Návrh architektury

Budou vytvořeny 4 možnosti přístupových práv, přesně jak to definuje schéma pro bankovní akreditiv v teoretické části. Každá strana bude mít přístup vždy k jedné větvi v definici API pro zapisování a validaci transakcí. Každá entita také dostane možnost vidět vždy aktuální stav ve, v němž se proces bankovního akreditivu nachází, a výpis všech dostupných dat v blockchainu. Při stavu dokončení celého procesu se může následně celý běžící blockchain odstranit a entity si mohou ponechat data pro další zpracování právě díky kopii výpisu všech proběhlých transakcí na blockchainu, uložené ve formátu JSON. Součástí komunikace v takto vytvořeném blockchainu budou pak i pomocná ověřování a zpětná kontrola všech transakcí v blockchainu, aby žádná strana nemohla komunikaci v rámci bankovního akreditivu pozměnit. Uložená data tak budou unikátní.



Obr. 9.1: Blokové schéma architektury

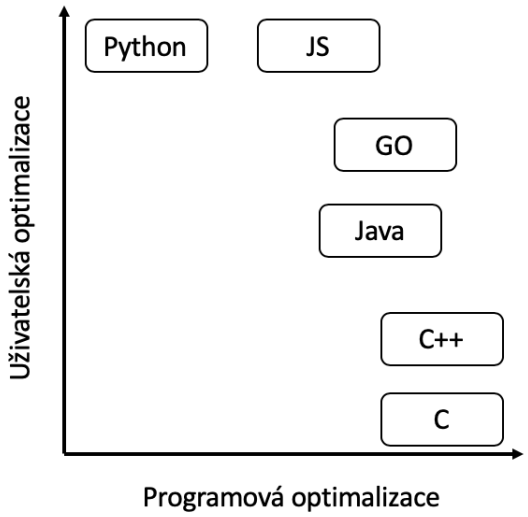
9.2 Programovací jazyk

Pro naprogramování jednotlivých funkčních bloků pro blockchain funkcionalitu, respektive chaincode, kde běží celá správa a programové jádro aplikace, slouží programovací jazyk GoLang. GoLang je programovací opensource jazyk vytvořený firmou Google, který má za cíl co nejnižší kompilační čas a jednoduchost používání. Byl vytvořen vývojáři, kteří se podíleli na vývoji programovacího jazyka NodeJS. Chtěli vytvořit lepší programovací jazyk pro servery, než je NodeJS. Golang přináší velmi vysokou rychlost kompilace a běhu programu jako C/C++ jazyky. Na druhou stranu je ale velmi uživatelsky přívětivý k vytváření kódů a přehledný. Kombinace těchto dvou přístupů nám přináší rozumný kompromis, a proto je tento jazyk zvolen k vypracování implantační části této diplomové práce.

Golang je výhodný pro použití hned z několika důvodů. Chyby v kódu jsou vždy odchyceny už při kompilaci kódu. Zavádí objektově orientované programování a dědičnost, i když je postaven na základech jazyka C. Zároveň je také stále velmi jednoduchý. Umožňuje implementaci různých knihoven. Samotný programovací jazyk byl vytvořen pro správu velkého objemu dat a je využíván firmami, které pracují s velkým tokem dat přes síť, jako jsou například Youtube, Netflix nebo Dropbox. Další výhoda oproti konkurenčním jazykům je rychlost a alokace paměti, kterou vykonává garbage collector při běhu programu. Golang také podporuje zavádění jednotkových testů a to také podporuje využívání tohoto jazyka pro složitější implementace. Stejně jako Java i Golang je nezávislý na platformě a může být skrze binární formu zkompilován všude. Hlavním důvodem, proč byl vybrán namísto Javy pro implementaci, je skutečnost, že blockchain implementace bude ve velké míře komunikovat se serverem a docílíme tím co nejnižšího času zpracování dat při běhu kódu.

Nevýhody při využití tohoto programovacího jazyka spočívají v opensource licenci, a to tedy že zde není žádná podpora, v případě problému se musí programátor spolehnout spíše na neoficiální fóra. Rozšíření v rámci knihoven není tak široké, protože musí mít rozdílnou strukturu, jako například knihovny pro jazyk C. V rámci implementace také neexistuje žádný oficiální nástroj pro vytváření uživatelského rozhraní a většinou se jedná o kompilaci na serveru.

Na obrázku 9.2 na další straně můžeme vidět uživatelskou a programovou optimalizaci v porovnání s jinými, nyní velmi používanými jazyky.



Obr. 9.2: Porovnání programovacích jazyků

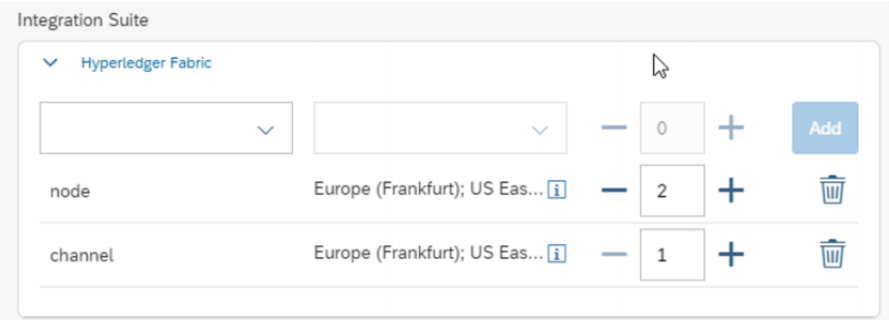
9.3 Základní konfigurace

Pro nastavení základní konfigurace a spuštění blockchain služeb na SAP Cloud Platform je potřeba vytvoření účtu. Účet je nutný nakonfigurovat na několika úrovních. Samotná registrace účtu a možnost práce na Hyperledger fabric je možná pouze na Cloud Foundry. Cloud foundry je vývojové prostředí pro škálovatelné web aplikace, které mohou být spuštěny na cloudu pod platformou SAP. Cloud foundry je open-source platforma, která funguje na systému PaaS a SAP aktivně rozšiřuje funkčnost této platformy. Pro spuštění Cloud foundry máme na výběr z mnoha poskytovatelů infrastruktury například Amazon AWS, Microsoft Azure nebo vlastní SAP datová centra. Podpora pro technologii blockchain je poskytována pouze na Amazon AWS, bude tedy vytvořen zde.

Globální účet

Při vytváření globálního účtu tedy volíme infrastrukturu, pro kterou bude vytvořen. Další krokem je přiřazení služeb. Pro naši implementaci budeme potřebovat samotnou Hyperledger fabric integraci a SAP HANA Service pro správu databáze. Přiřazením těchto služeb v dalším kroku konfigurace následně přiřadíme množství jednotlivých instancí, které budeme využívat pro implementaci. Na obrázku 9.3 níže vidíme přiřazení pro všechny služby, které budeme potřebovat - 2 nody, 1 kanál a 1 instanci pro SAP HANA databázi, ke které se následně při pozdějším vytváření

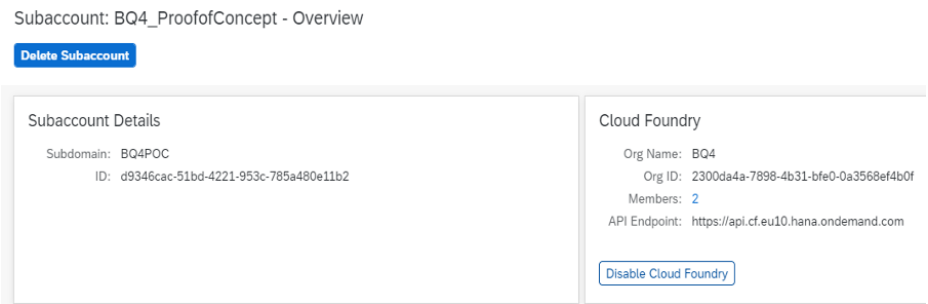
přiřadí ještě velikost, již bude databáze využívat. Posledním krokem je potom už jen kontrola zvolených údajů a vytvoření globálního účtu.



Obr. 9.3: Konfigurace globální účtu

Sub-účet

Předchozími kroky jsme vytvořili globální účet na serveru Europe (Frankfurt). Zde se pak už jen následně kliknutím na Open in cockpit dostaneme do prostoru, kde budeme konfigurovat vlastnosti a chaincode. Dalším krokem ke zprovoznění je vytvoření sub-účtu. V jeho rámci můžeme přiřazovat jednotlivé služby z globálního účtu, který může sloužit pro vývoj více lidem, je možné dokonce pracovat na několika projektech zároveň. Tento způsob také umožňuje kontrolu, jaké služby jsou využívány. V rámci naší implementace jsem tedy vytvořil sub-účet BQ4 ProofofConcept. Zde je následně ještě nutné zavedení využívání služby cloud foundry, kde se vytvoří API end point a organizace, do níž můžeme přidávat účastníky, kteří budou mít možnost editovat nebo číst konfiguraci pro tento sub-účet. Na obrázku 9.3 níže vidíme hlavní obrazovku s konfigurací sub účtu po zapnutí Cloud foundry. Součásti



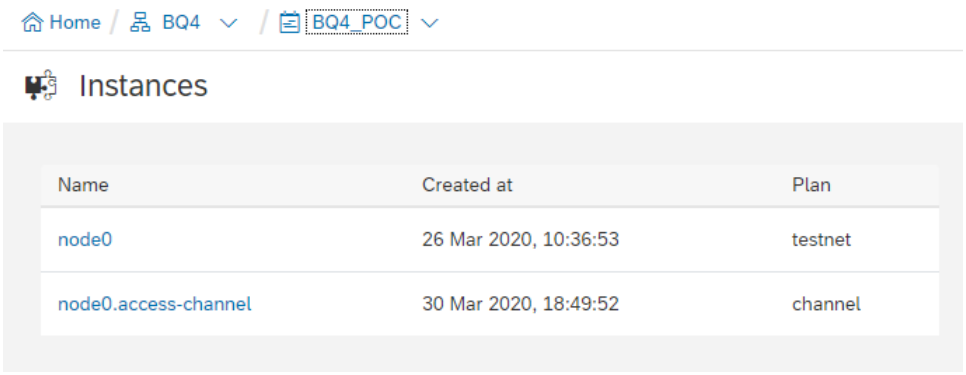
Obr. 9.4: Konfigurace sub účtu

nastavení sub-účtu jsou také alokace instancí, které může tento účet využívat. Opět zde můžeme nastavit a limitovat množství přidělených prostředků a limitovat tak

zbytečně blokované prostředky pro jiný vývoj, za než se bude jinak platit dle modelu placení jen za využívané instance.

Prostor

V rámci sub-účtů musíme ještě nakonfigurovat samotný prostor nazývaný jako space. Ten slouží k vytváření už samotných instancí jednotlivých služeb. Zde si v obchodě služeb vybereme právě instanci hyperledger fabric. Z ní vytvoříme 2 samostatné entity. Jedna bude kanál, který bude sloužit jako přístupový bod do blockchainu. Následně k tomu ještě potřebujeme vytvořit samotnou reprezentaci blockchainu, to bude tedy instance testnet node, která umožňuje vytváření a testování multi platformní aplikace. Tu nelze využít na produktivním kódu, ale to na implementaci nyní nemá vliv. Pro produkční kód by se jinak volila možnost dev, která mnohem má širší možnost implementací a přístupů. Dev verze je ale také samozřejmě dražší, proto se pro vývoj konceptů vždy využívá testnet verze. Poslední nutná instance je samotná implementace SAP HANA služeb pro alokaci paměti na serveru. Na obrázku 9.3 níže vidíme seznam všech instancí, které byly vytvořeny.



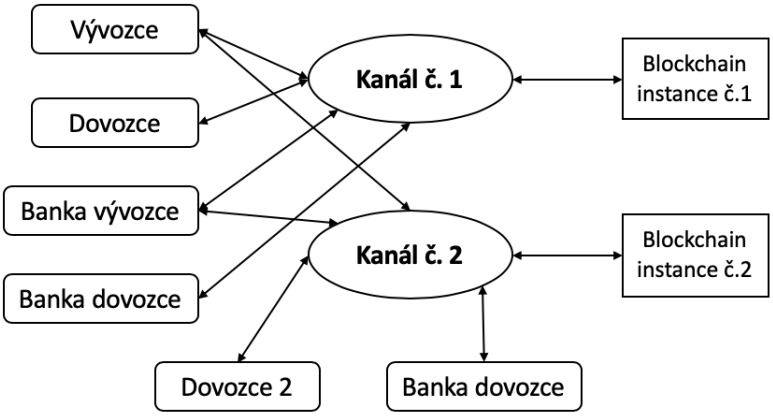
Obr. 9.5: Aktivované instance v prostoru BQ4 POC

Vazby a přístup do hyperledger fabric

Pro přístup do samotné konfigurace jednotlivých entit blockchainu se musíme připojit skrze vytvořený prostor. Uvnitř něj můžeme vytvořit nody a služby blockchain. Jednotlivý uzel vždy reprezentuje jednu instanci, kterou vytvoříme a vložíme do ní náš chaincode (programovou nadstavbu). Tyto nody také reprezentují blockchain síť. Pro implementaci budeme vytvářet tedy jeden hlavní a jeden záložní uzel pro ukládání komunikace v bankovním akreditivu. Z důvodu, že nebudeme vytvářet pro každou stranu jednotlivé nody, musíme vytvořit kanál, pomocí kterého se budou účastníci připojovat, zapisovat a číst data uložená v blockchainu. Kanál slouží pro

připojení a ověření jednotlivých účastníků do blockchain sítě. V rámci kanálů může mít uživatel hned několik aktivních připojení. Skrze kanál bude dostupný přístup k nodu, který uchovává transakční data pouze po splnění ověřovacích pravidel a vygenerováním klíčů pro jednotlivé entity.

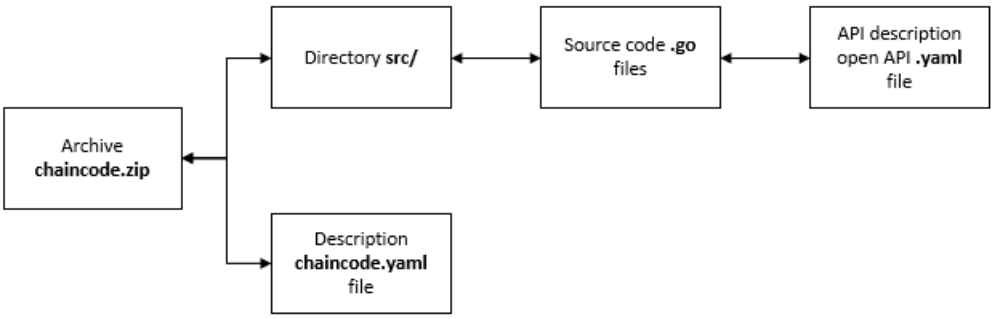
Na obrázku 9.3 níže vidíme reprezentaci jednotlivých nodů a uživatelů, kteří se do různých blockchainů můžou zapojovat nezávisle. V rámci jednoho kanálu máme připojené 4 entity a následně každá z nich může být ještě navíc připojena do jiných sítí. Tímto způsobem je vyřešen problém s ochranou osobních dat, který byl zmíněn v teoretické části jako jedna z možných limitací práci s blockchainem. Další výhoda tohoto použití spočívá také v tom, že pokud jedna z entit neustále udržuje vztah a například často vzájemně obchodují, není pak už nucené znovu ověřování pro vstup do blockchainu, ale stačí pouze obnovení již předešlého a nezrušeného spojení pro obnovení komunikace. Tímto způsobem velmi usnadníme znovunavázání komunikace mezi účastníky, kteří velmi často obchodují. Pro strany, které spolu nekomunikují tak často, dosáhneme tak možnosti vytvořit velmi důvěrnou komunikaci díky ověřování platností transakcí dalšími 2 mezistranami, které chtějí dohlížet na dokončení všech kroků.



Obr. 9.6: Přístup do blockchain instancí

9.4 Chaincode

Chaincode je skupina souborů, které mají na starost všechny interakce s námi zvolenou platformou, tedy SAP Hyperledger Fabric. Chaincode je v našem případě napsán tak, aby umožňoval číst a zapisovat data do blockchainu a také aby bral do uvážení námi nadefinované vstupní podmínky a zajistil tak unikátnost zápisů do blockchainu.



Obr. 9.7: Chaincode struktura

Na chaincode se můžeme podívat ze dvou perspektiv. První - jako vývojář a druhá jako správce sítě blockchainu. Správce má na starost připojené uživatele a blockchain jako celek. Zde jsme pak schopni provádět úpravy pomocí verzování chaincodu. Vývojář pak napíše kód v programovacím jazyce GoLang, kde nadefinuje základní funkce, jež jsou pak přístupné přes API v rámci sdíleného chaincodu. Všechny takto vytvořené soubory se sbalí do .zip souboru a umístí na platformu, která sama následně zajistí rozbalení a zprovoznění.

Soubor chaincode.yaml

Tento soubor má na starost pouze verzování na SAP Hyperledger Fabric platformě a to tak, že je zde jen vloženo číslo verze, které odpovídá aktuální verzi nacházející se v adresáři /src. Vždy se ale musí označit názvem námi vytvořeného kanálu a verzí, nic jiného se zde nepíše. Jde o soubor pro správce sítě.

Soubor LC.go

V souboru LC.go inicializujeme a programujeme funkce, které potřebujeme na vytvoření blockchainu. Zde budeme programovat všechny možné práce s daty, jež budou blockchainu následně zpracovávat a ukládat. V rámci diplomové práce bude naprogramována inicializační funkce a následně i funkce pro čtení a zápis, která bude mít ošetřeny základní možné chyby s návratovými kódy, které může uživatel při zadávání dat do blockchainu udělat.

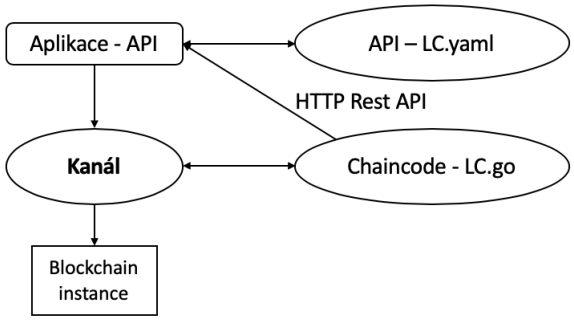
Soubor LC.yaml

V souboru LC.yaml definujeme vstupní parametry pro vytvořené funkce v předchozím souboru. Součástí těchto definicí jsou i návratové kódy, které dostane v případě

úspěšného, nebo neúspěšného běhu funkce. Můžeme i nastavit povinné, či nepovinné parametry pro vstupní hodnoty jednotlivých funkcí.

9.5 Práce s API

Pro komunikaci mezi uživatelem a samotným kódem budeme využívat API, které si vytvoříme pomocí nástroje swagger. Na obrázku 9.5 níže vidíme reprezentaci, jak funguje vazba mezi samotnou instancí blockchainu, kanálem a API, s nimiž komunikuje uživatel.



Obr. 9.8: Fungování API

Swagger

K vytvoření API využijeme opensource framework Swagger. Jde o editor a rozhraní pro vytváření a testování API. V rámci implementace jsou vytvořeny 4 různé sady API, aby každá entita mohla provádět jen určité operace nad blockchainem. Na přiloženém obrázku 9.5 můžeme vidět, jakým způsobem jsou API vytvořeny. Pro každou stranu jsou vytvořeny různé akce POST, případně GET, které slouží k zajištění kontroly nad celým procesem bankovního akreditivu. V rámci těchto metod máme doplněny i povinné, případně nepovinné parametry, jež musí být zadány z důvodu konzistence dat. V rámci každé metody je vytvořen specifický typ objektu JSON, který ukládá data o aktuálním bankovním akreditivu. Objekt typu JSON se pak následně dá velmi lehce zpracovat pro napojení jakýchkoliv dalších služeb.

Obr. 9.9: Vytvořené API

Autorizace přístupu při použití API

V rámci kontroly autorizace pro použití je možné v samotném swagger editoru vyzkoušet funkčnost jednotlivých metod, zda je vidět, jaká data posílají do blockchainu v rámci rozhraní API. Na tomto místě také definujeme ověření pro entity, a proto před každým zápisem se strana musí ověření stisknutím na tlačítko authorize a to garantuje možnost využívat API metody.

Vyvolání, dotazy na HTTP požadavky

Předchozími kroky jsme tedy provedli prvotní implementaci všech nutných prvků. Nyní zbývá nakonfigurovat jejich funkčnost. Nadefinované API vytváříme tedy pomocí nástroje swagger. Všechny vytvořené metody a funkce uvnitř souboru LC.yaml musí být správně namapovány na http požadavky. HTTP nám umožňuje provádět základní CR operace, tedy – vytvořit, číst. V implementaci, která je součástí diplomové práce, budeme tedy využívat jen zápis a čtení. Součástí hyperledeger fabric jsou k dispozici funkce k vyvolání tohoto požadavku, jako je například create. Druhým způsobem práce s daty je dotaz na požadavek, a to ve formě funkce čtení z blockchainu.

Volání

Pro volání a používání metod POST a GET je nutné definovat povinné parametry. Hlavním parametrem je operationId. Další definice jsou pak už pro samotné textové pole. Definujeme ID, které slouží pro identifikaci pro další zpracování, a můžeme také vidět, jaká strana tuto funkci použila v rámci metod v souboru LC.go. Dalším parametrem jsou samotné JSON objekty, které očekáváme na vstupu. Samotné objekty jsou definovány v úvodu souboru LC.yaml, kde jsou předdefinovány maximální

délky a datové typy jednotlivých polí. Na hodnoty vstupu jsou vždy před uložením do blockchainu prováděny kontroly pro zachování co nejvyšší možné konzistence dat.

V rámci návratových kódů nejen definujeme samotný kód, ale může být využito informační pole k napsání informačního textu pro uživatele, aby bylo označeno, ke které chybě došlo. V rámci této implementace zajišťujeme unikátní transakce a vstup správných parametrů.

```
135    post:
136      tags:
137      - "Exporter"
138      operationId: vytvor_exporter
139      summary: Krok č.1 a č.5
140      consumes:
141      - application/json
142      parameters:
143      - $ref: '#/parameters/objId'
144      - $ref: '#/parameters/objJSONexport'
145      responses:
146        201:
147          description: Transakce uložena.
148        400:
149          description: Špatně zadané parametry.
150        409:
151          description: Tento záznam již existuje.
```

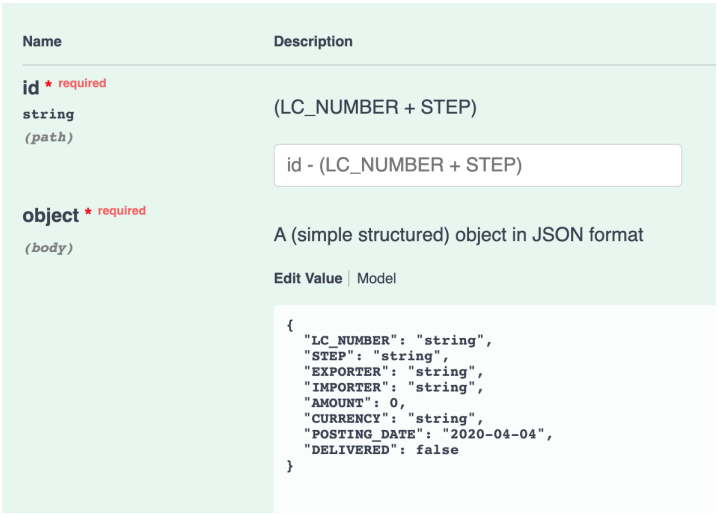
Obr. 9.10: Programová definice API parametrů

Na obrázku 9.5 výše vidíme všechny parametry pro vytvoření funkce POST pro vývozce i s konfigurací návratových kódů.

Definice cest pro chaincode

V rámci konfigurace chaincodu definujeme paths neboli cesty jednotlivých metod k nám vytvořeným entitám, které bude do blockchainu zapisovat. Jelikož budeme vytvářet 4 různé entity s přístupem do blockchainu, budou vytvořeny i 4 cesty, ve kterých jsou vytvořeny i metody. Každá entita má tak jiné metody a parametry, které budou zapisovány do blockchainu. Na obrázku 9.5 níže vidíme výpis metody post pro dovozce, kde jsou definované i parametry a nápověda, v jaké formě mají být jednotlivá data vložena. Nejdůležitější je id, která se skládá z čísla bankovního akreditivu a kroku, který entita má aktuálně vykonat .

V rámci definice jednotlivých uživatelů, kteří jsou označení, můžeme také měnit datové typy jednotlivých bloků a následně v souboru LC.go přidávat různé validace při běhu a kontrolu konzistence běhu programu, aby například nedošlo k přeskočení nějakého kroku v rámci bankovního akreditivu. Tato kontrola je poskytnuta i poslední metodou list, která vypíše všechny transakce uložené v aktuálním blockchainu, a je pak jednoduché dohledat, v jaké fázi se bankovní akreditiv nachází.



Obr. 9.11: Vstupní parametry pro API

Konfigurace

Součástí spuštění funkčního kódu pro chaincode je také nutná konfigurace návratových kódů. Pro každou metodu, kterou voláme musíme mít nadefinované návratové kódy. Nejčastěji se využívají kódy 200-299, které signalizují, že došlo k úspěšnému zapsání nebo zobrazení. Pro neúspěch se nejčastěji využívají kódy 400-499. V případě nutnosti je možné vytvořit různé návratové kódy, které mohou opět měnit i obsah a hlášky v závislosti na tom, jakým způsobem bude na tyto kódy reagováno v definici samotných metod, tedy v souboru LC.go. Poslední část, která se definuje uvnitř tohoto souboru, je popis API, které využíváme a jež se zobrazí vždy při zápisu do blockchainu. Nejčastěji se zde implementuje vyrozumění pro práva a možnosti využití definovaných API.

Validace

Validace funkčního programu pro správu API volání a vyvolání lze provést několika možnými způsoby. Všechny tyto kontroly zajistí konzistenci dat a optimalizaci programu. V rámci implementace nemůžeme ověřit správné vyvolání go kódu, protože http request nevkládáme do implementace blockchainu na SAP Cloud Platform, ale tato kontrola je možná zkompilováním kódu LC.go, kde se tyto vazby ověřují.

Validace swagger

První krokem je možná kontrola přímo ve webovém editoru swagger. Aplikace provádí všechny kroky, které se vykonají až po samotný http požadavek, kterým se následně volá funkční kód go. K dispozici máme i response body, jež obdržíme při

vyvolání jednotlivého příkazu POST a jaké data se budou následně do blockchainu ukládat. V rámci GET dostaneme pouze chybový stav, protože žádná transakce umístěná pomocí metody POST se do samotného blockchainu nezapíše. Jde tedy spíše o základní funkční test.

Validace během kompilace

Dalším krokem validace funkčního kódu je při generaci funkčního kódu z konzole go – tomu se bude věnovat příští kapitola. Při samotném generování a kompilaci programu je kontrolována syntax i přiloženého souboru LC.yaml a jsou protestovány všechny metody, zda mají správně adresované parametry a odpovídají tak i těm, jež jsou definovány v souboru LC.go.

Validace na SAP Cloud Platform

Nejdůležitější validace funkčnosti a obou programových částí probíhá pak už samotným vyzkoušením zápisu do blockchainu přímo na SAP Cloud Platform. Po nahrání konfiguračních a programových souborů může být provedena kontrola funkčnosti všech metod. Nyní už i pro metody POST a GET budeme kontrolovat data již zadaná do blockchainu. Tento test už je jen potvrzením samotné funkčnosti kódu. Všechny předchozí kontroly jsou dostatečné pro odhalení chyby.

Validace pomocí verzování

Poslední možnost kontroly našeho kódu je také při verzování a měnění obsahu jednotlivých souborů. Při nahrávání na SAP Cloud Platform musí být všechna data v hlavičce a těle programu správná. Toto zajistí, že uživatel opravdu po změně na jiný chaincode dostane funkční verzi.

9.6 Programová logika

Programová logika a definice všech tříd je uložená v konfiguračním souboru LC.go. Hlavní prerekvizitou před samotným psaním kódu je příprava vývojových nástrojů. Prvním krokem je stažení Hyperledger fabric balíčků. Nainstalujeme tedy tři balíčky a už můžeme provádět syntax checky jen napsáním do příkazové řádky go build a přidáním cesty k souboru. Samotný vývoj probíhá již ve zmíněném souboru.

Hlavička programu

Program začíná implementace balíčku main, který slouží jako hlavní spustitelný soubor pro příkaz go build, kterým můžeme ověřit spustitelnost z konzole. Dalším

krokem je implementace pro import nainstalovaných knihoven. Dále už přichází definice samotného programu, a to vytvoření typu Chaincode a spuštění rozhraní pro tuto entitu pomocí příkazu shim.Start.

První funkce Init zajišťuje správnou inicializaci a vytvoření blockchain instance a je volána pouze na začátku. Druhá funkce Invoke implementuje jednotlivé funkční bloky, které budou využívat metody. V případě této implementace jsou vytvořeny funkce jako read, write, list, validate, success a error v mnoha variantách pro každou entitu. V případě jiných funkcí je vrácen error, a je tedy ošetřena chyba během implementace nevložených funkcí a volání skrze API.

Funkční bloky

Samotné funkční bloky definujeme jako funkce, kde pro read nebo write máme vždy nadefinované vstupní parametry. První je instance blockchainu vytvořená během funkce Init a parametr, který se bude zapisovat nebo číst z blockchainu. Součástí této metody je i zajištění konzistence dat, a to díky kontrole na všechny vyplněné parametry. Tímto způsobem budou muset být nakonfigurovány všechny funkce read a write, které bude program využívat. Bude jich tedy celkem 8. Toto nám bude sloužit jako připravené rozhraní k napojení na vytvořené API v předchozí kapitole.

Funkce write

Fungování funkce write je založeno na zápisu vstupních dat vložených uživatelem. Před každým vstupem je provedena kontrola jejich konzistence. Následně přijatá data uložíme do proměnných pro samotné zpracování a kontroly v rámci GO programu. Po dokončení interní logiky programu už zbývá jen data odeslat do blockchainu a k tomu nám slouží PutState metoda. Tato metoda má vstupní parametr ID, které bude sloužit i k vyhledávání, a také pole dalších textových prvků, jež budeme vkládat do blockchainu. Programovací jazyk GO tato data ale nejdříve z řetězce konvertuje do bytového pole pro jejich rychlejší správu. Posledním krokem je už jen návrat zprávy, který signalizuje úspěšné zapsání do blockchainu, která opět navázáním na API vrátí uživateli hlášku úspěchu ze souboru LC.yaml. Všechna data jsou uložena ve formátu JSON objektu.

Funkce read

Funkce read je mnohem jednodušší, protože na vstupu vždy přijme pouze jeden vstup, a to ID. Po provedené asociaci proměnné v rámci funkce je zavolána metoda GetState. Ta potom už jen vrátí z blockchainu vyhledávaný blok v případě, že existuje, pokud ne, tak vrátí error, že záznam nebyl nalezen.

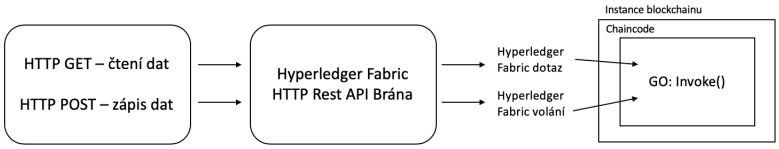
Funkce validate

Funkce validate má na starost ověření vstupních dat, která se následně uloží do blockchainu. Součástí implementace je kontrola délky a datových typů k zachování konzistence dat ukládaných do blockchainu.

Funkce list

Funkce list je definována pro všechny účastníky stejně a poskytuje výpis všech transakcí uložených do blockchainu podle ID. Každý uživatel pak pomocí ID může pomocí vlastní metody get získat další parametry uložené v transakčním bloku.

Architektura volání API z funkcí



Obr. 9.12: Proces volání funkcí

Na obrázku 9.6 výše vidíme způsob napojení funkcí pomocí softwarového balíčku pro vývoj přes bránu, kterou poskytuje samotná SAP Cloud Platform k samotným API voláním, která jsou nakonfigurována v souboru LC.yaml. Vytvořením požadavků a voláním se vždy spustí funkce invoke a pro nadefinovanou akci bude vrácena návratová hodnota odpovídající výsledku požadavku. Do takto vytvořeného souboru už můžeme jen pomocí komentářů vložit popisové informace. Soubor následně uložíme jako LC.go.

9.7 Umístění na SCP

Jakmile dokončíme veškerý vývoj, je nutné zabalit všechny soubory do ZIP formátu a nahrát je na samotnou SAP Cloud Platform. Nutný je jen požadavek, aby ZIP soubor měl stejný název jako yaml soubor, kde jsou nakonfigurována API volání. Důležitá je také samotná struktura, jak budou jednotlivé soubory uloženy na SAP Cloud Platform. Struktura je naznačena na níže přiloženém obsahu CD.

```
/ ..... kořenový adresář
├── /src ..... složka src
│   ├── LC.go
│   └── LC.yaml
├── LC.zip ..... zip soubor pro verzování
│   ├── /src
│   │   ├── LC.go
│   │   └── LC.yaml
│   └── chaincode.yaml
└── chaincode.yaml ..... informace o verzi
```

Instalace

Instalace samotného obsahu těchto konfiguračních a programových souborů probíhá na SAP Cloud Platform v sekci chaincode pro nastavení nodu. Instalace se provádí jednoduchým otevřením vyskakovacího okna pro instalaci a následným přetažením ZIP souboru a potvrzení o nahrání tohoto soubor. Rozběhnutí a spuštění chaincodu provedeme klinutím na tlačítko instantiate, které uvede daný chaincode do provozu.

Servisní klíče

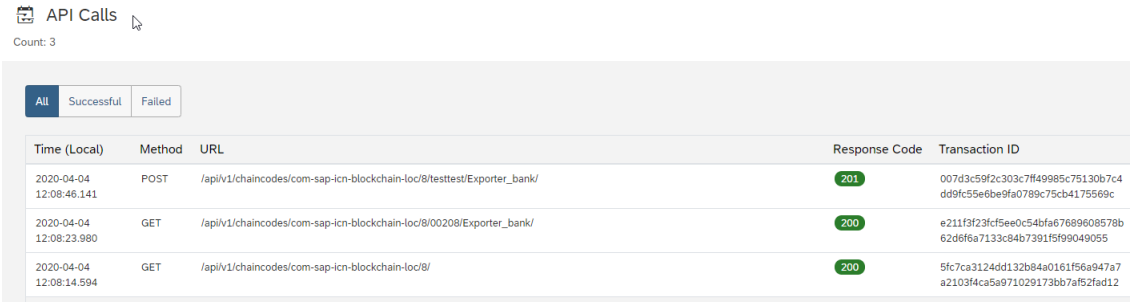
Pro umožnění testování vytvořených API skrz prohlížeč je nutné vytvoření korespondujících klíčů pro entity. Servisní klíč vytvoříme na úrovni kanálu, kde v sekci Service keys vygenerujeme clientID a clientSecret. Tyto údaje pak slouží k ověření při kliknutí na tlačítko autorizace při testování implementace. Toto ověření bude nutné pro každou entitu, aby mohly zapisovat pouze určité bloky během procesu bankovního akreditivu. Každá strana si tedy vygeneruje jen klíče opravňující ke správě jen jejich metod a funkce list.

9.8 Práce s SAP Hyperleger fabric workspace

Uživatelské rozhraní pro blockchain

Pro jednodušší orientaci a správu dat ve vytvořeném blockchainu existuje uživatelské rozhraní na SAP Cloud Platform. Vytvořením prostoru na SCP dostaneme nabídku všech Cloud služeb, které SAP nabízí. Zde si vybereme Hyperledger fabric. Poté vytvoříme instanci blockchainu - node0. Uvnitř dashboardu nyní máme přístup do mnoha kategorií: Node, Network, Channels, Certificate Authority a Logs. Například v poli Logs můžeme vidět časy a hash hodnoty volaných prvků nad naší databází. Naopak v položce channels se dostaneme do rozhraní pro správu samotného blockchainu a máme zde možnost pomocí lišty explore procházet bloky, upravovat verze přes záložku chaincode. Můžeme zde také vidět samotné volání API, které

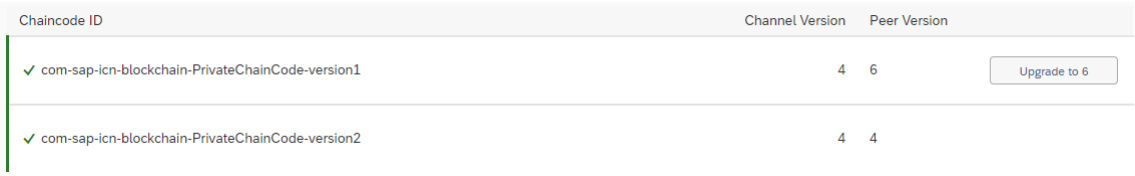
uživatel použil. Na spodním okraji obrazovky je pak uveden odkaz na dokumentaci k samotné terminologii nebo i podpora pro vytváření modifikací blockchainu skrz uživatelské rozhraní.



Obr. 9.13: Kontrola volání API

Změny v kódu pomocí verzování

Změny v kódu se zde provádí velmi jednoduše. Přepíšeme aktuální kód, soubory komprimuje a následně je vložíme do struktury ve stejném formátu, jak je potřeba pro první upload v záložce chaincode. Jedinou změnou pak tedy bude hodnota v proměnné verze v souboru chaincode.yaml. SAP Cloud Platfrom pak již rozezná jiné verze při uploadu na Cloud Platform a provede aktualizaci všech souborů a bude tím vytvořena změna. Tyto změny dle verze můžeme sledovat i z uživatelského rozhraní, takže jsem schopni přehrávat jiné verze a snadno zjistit, jaká je aktuální při testování. Na přiloženém obrázku 9.8 můžeme vidět, že uživatelské rozhraní dokonce informuje o novější verzi pro blockchain a můžeme aktualizovat na vyšší, v tomto případě na verzi 6.

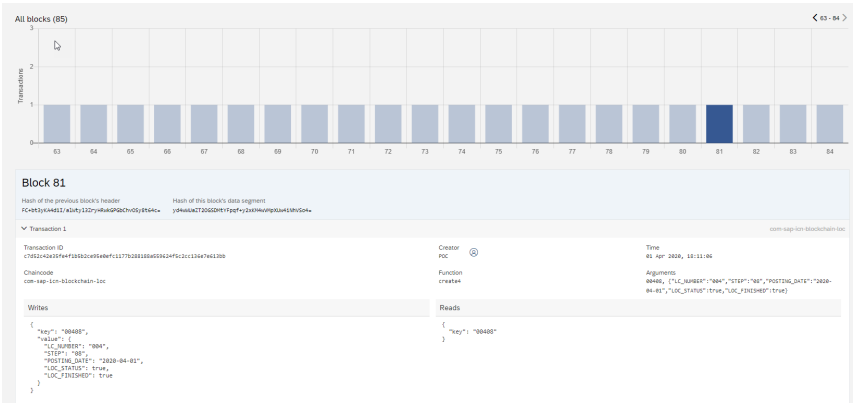


Obr. 9.14: Verzování zdrojového kódu

Vlastnosti bloků

Pro Hyperledger Fabric je vytvořen přehledný Dashboard. Jde o aplikační rozhraní na procházení databáze. Každá transakce je zobrazen v přehledném uživatelském rozhraní, kde můžeme procházet jednotlivé bloky v databázi. Jak vidíme na obrázku 9.8 pod tímto textem, rozkliknutím jednotlivého bloku dostaneme více informací,

které mohou účastníci vidět, a ověřit tak obsah transakcí v blockchainu. Vidíme zde i časové razítko, kdy byl blok vytvořen, jeho hashové hodnoty a další parametry.



Obr. 9.15: Blockchain explorer

9.9 Implementace bankovního akreditivu

Implementační část této diplomové práce se zabývá správou procesu pro bankovní akreditiv. Bankovní akreditiv byl popsán v teoretické části této práce. Součástí implementace je tedy vytvoření aplikačního rozhraní pro všechny čtyři entity, které jsou účastníky bankovního akreditivu a nějakým způsobem musí interagovat navzájem.

Aktuální řešení problému

Nyní se správa pro bankovní akreditiv provádí pomocí komunikační sítě SWIFT. Konkrétně pro výměnu informaci pro jádro bankovního akreditivu slouží varianta SWIFT 798. Pro podpůrnou komunikaci mezi klientem a bankou se pak používá ve verzi 700. Jde tedy o zprávy, které se nejčastěji týkají komunikace klienta a banky. Implementace v rámci této diplomové práce bude vytvořena v koncepci tohoto režimu pro simulaci tohoto procesu, a nebude tedy zpracovávat data pro formáty SWIFT. Toto rozšíření by bylo možné, a to konfigurací v souboru LC.go a konverze dat z JSON objektu.

4 účastnící – 4 přístupy

Pro správu bankovního akreditivu je potřeba vytvoření metod pro zápis a čtení obsahu jednotlivých bloků s tím, že pro zápis mohou využít jen své autentické zprávy.

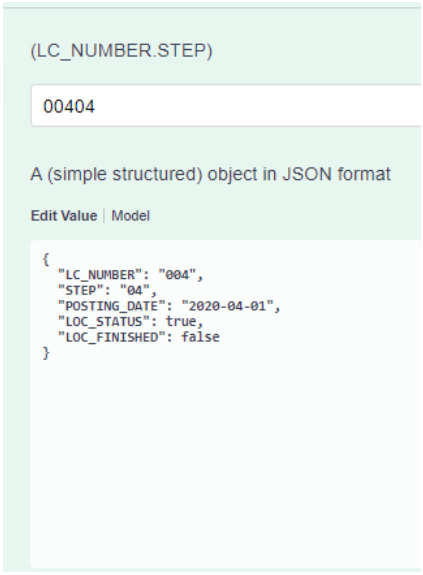
Entita vývozce bude mít na starost inicializaci blockchainu a poslání první zprávy dovozci o připojení do blockchainu. Dále bude také informovat v rámci pátého kroku

bankovního akreditivu o splnění závazku a následně předání potvrzení bance vývozce o dokončení procesu.

Dovozce bude disponovat pouze žádostí na svoji banku pro vytvoření bankovního akreditivu mezi bankami a následně bude mít možnost kdykoliv nahlížet na aktuální stav bankovního akreditivu díky využití možností vidět informace o všech blocích v blockchainu. Dále také získá funkci zápisu v rámci kroku číslo 6, kdy potvrdí splnění závazku, a následně dojde k uvolnění odměny za plnění.

Banka vývozce má na starosti informování o vzniklém bankovním akreditivu dovozce a také prezentace všech potřebných informací o plnění závazku k bance dovozce.

Banka dovozce má na starost také pouze informování a schvalování jednotlivých dokumentů a předávání informací bance vývozce. Alternativní možnost inicializace blockchainu může provádět i dovozce. V tomto případě dojde jen k přeskočení prvního kroku a následné číslování i postup bude pořád stejný. Dle obrázku 9.1 jsme v rámci úvodu praktické části vytvořili tedy 4 možné přístupy pro klíčové účastníky komunikace v rámci bankovního akreditivu, kteří jsou schopni informovat o obsahu v jaké se aktuálně nachází. Všechny tyto funkce pro čtení a zápis jsou implantovány a jsou rozvrženy v souboru LC.yaml a LC.go. Zde je pak autentizace pro jednotlivé entity zajištěná vygenerováním servisních klíčů a následně tak mohou entity přistupovat pouze ke svým funkcím write. Na obrázku 9.9 níže vidíme vzor zápisu pro funkci write ve 4. kroku.



Obr. 9.16: Obsah funkce write pro 4. krok

Aplikační průběh celého procesu

Pro kontrolu celého procesu bankovního akreditivu jsou implementovány kontroly na úrovni aplikační a také interní se závislostmi na zajištění konzistence zapisovaných dat. První kontroly probíhají už při zadání dat, byly zadefinovány přesné datové typy pro jednotlivá pole, tak aby nedošlo ke vzniku nesmyslných dat. Tato kontrola zamezí zápisu do blockchainu v případě, že jsou vložena špatná data a uživatel je může hned opravit. Po správném vstupu dat skrze uživatelské rozhraní jsou všechna data zpracována a editována zdrojovým souborem pro chaincode jednotlivé entity blockchainu.

Inicializace a ukončení procesu

Komunikace skrz blockchain pro správu bankovního akreditivu začíná vždy tím, že vývozce, případně dovozce zažádá poskytovatele blockchain sítě o rezervaci prostředků na vytvoření blockchain sítě skrz SAP Cloud Platform. Zde následně vývozce, nebo dovozce odešle žádost pro druhou stranu, aby se do takto vytvořené sítě přidal na základě vygenerovaného klíče pro přístup do kanálu. Dalším krokem je pozvání bank obou stran, aby byly všechny strany pro kontrolu stavu bankovního akreditivu uvnitř. Nyní může začít celý běh bankovního akreditivu. Jakmile proběhnou všechny kroky a závazky jsou naplněny, může vývozce ukončit běh blockchainu a vytvořit si pro nutnost kopii pro vlastní účely. V případě, že strany budou spolu obchodovat často, je možné tuto blockchain síť ponechat a přidávat data o dalších obchodech. Do takto vytvořeného blockchainu můžeme vkládat i jiné bankovní akreditivy, a to díky dostatečné implantaci vstupních dat pro rozlišení dat mezi transakcemi. Každá transakce je vždy, ale ukončena záznamem vývozce banky, kdy je proměnná LOC FINISHED nastavena na boolean hodnotu TRUE.

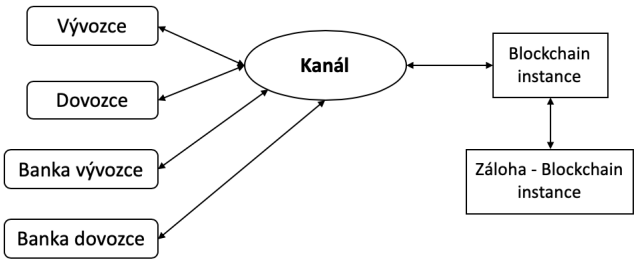
Jeden z požadavků při vytváření této praktické části bylo i snížení ceny pro správy tohoto procesu v blockchainu, kde by při využití klasického přístupu byla nutná jedna kopie blockchainu pro každou stranu. Z tohoto důvodu se pro správu a kontrolu dat používá pouze jedna tato instance. Z důvodu větší bezpečnosti a zálohování dat byl proto vytvořen záložní blockchain, který je popsán v kapitole níže. Využití pouze jednoho nodu je možné, ale v produkčním řešení se musí počítat i s vytvořením záložního nodu pro zajištění vyšší bezpečnosti dat v případě útoku na dostupnost služeb.

Správný obsah pro dokončení bankovního akreditivu je definován v přiloženém word dokumentu, který je také umístěn na CD společně s programovou částí této implementace. Dodržování hodnot kontrolních proměnných je důležité, aby ostatní účastnící věděli, v jakém stavu momentálně bankovní akreditiv je.

Záložní uzel

Z důvodu požadavku pro snížení ceny implementace bylo nutné limitovat počet uzlu vytvořených pro záznam historie transakcí. Jakýkoliv další uzel přidaný do sítě zvyšuje transparentnost a integritu zvlášť ve veřejném blockchainu a to je v protikladu s požadavkem na bezpečnou implementaci jeho technologie. Implementace v rámci diplomové práce byla založená na předpokladu privátního blockchainu, a proto za účelem snížení ceny bude využíván pouze jeden uzel, do něhož budou uživatelé přistupovat přes nadefinovaný kanál, který jim následně skrz autorizace zpřístupní všechny nutné metody a zprávy pro nutné transakce, které bude jednotlivá strana vyžadovat pro průchod procesu bankovního akreditivu. Takto vytvořená implementace se neliší příliš od sdílené centralizované databáze na serveru, která by mohla být využita obdobně. Implementace blockchain nám v tomto případě zajistí nemožnost měnit jednotlivé záznamy co implementace na vzdáleném serveru neposkytuje.

Klíčový rozdíl bude ten, že k implementaci bude vytvořen i jeden záložní blockchain instance, které bude uchovávat kopii tohoto blockchainu a bude vždy možné ověřit v porovnání se zálohovanou instancí, zda jsou všechny bloky vytvořeny se stejnou hash hodnotou, jako je uložená v instanci pro aktuální bankovní akreditiv. Na obrázku 9.9 níže vidíme grafickou reprezentaci pro vytvoření používané hlavní entity blockchainu a také záložního uzlu, ke kterému mají přístup 4 uživatelé.



Obr. 9.17: Grafická reprezentace záložního uzlu

Konečné zpracování dat

V rámci implementace je i požadavek na možnost uložení dat pro kopii i u samotných účastníků bankovního akreditivu. Po každém kroku write, read nebo list můžeme data, která se zobrazí ve formátu JSON, uložit na disk pomocí tlačítka download. Tato možnost docílí toho, že po dokončení procesu může být blockchain deaktivován a nemusí se platit za běžící instance na serverech.

9.10 Komplikace a dodatky pro implementaci

Během implementace vzešlo několik otázek pro správnou volbu implementace, a to nejen z hlediska programového, ale také procesní hledisko. Jakým způsobem jde o inovační nebo pokrokový systém pro řešení implementaci bankovního akreditivu při použití blockchain technologie?

Osobní data

První překážkou při vývoji bylo hned uchovávání osobních údajů, které jsou nově pod kontrolou Evropské unie na základě GDPR. Tato implementace je založena na privátním blockchainu, možná alternativní implementace by byla neuchovávat samotná data o jednotlivých bankovních akreditivech přímo v blockchainu, ale uchovávat zde jen hash hodnoty jednotlivých bloků s tím, že samotná data by byla přenášena zabezpečeným kanálem tak, aby nikdo nemohl tato data získat. Tato implementace se dá uvažovat jako možné rozšíření této implementace například při reálném používání v produkčním kódu při zajištění ještě vyšší bezpečnosti na základě požadavku potenciálního zákazníka.

Samotná definice SWIFT zpráv v případě využití migrace do toho formátu nenarušuje požadavky pro GDPR, protože se předávají informace pouze o jednotlivých obchodech, a nikoliv informace fyzických osob.

Volba technologie

Pro volbu technologie této implantace byl zvolen Hyperledger fabric, a to z důvodu snadného způsobu rozšiřitelnosti a jednoduchosti správy sítě. Pro vytvoření jednoduchých implementací, které budou sloužit pouze k zápisu bez jakýchkoliv kontrol nebo dalších parametrů při zápisu, se dá použít třeba i multichain. Další varianta byla i Quorum, které je také nabízeno na SAP Cloud Platform pro implementaci, ale nebylo zvoleno z důvodu velmi nové implementace pro SAP Cloud Platform a vyšší ceny na její provoz.

Rozšiřitelnost pro ostatní účastníky akreditivu

Požadavek na možnost rozšiřitelnosti také vznikl v době vývoje této implementace. Aktuální řešení funguje tak, že vývozce působí jako startovní bod a inicializuje blockchain síť, kam následně zve další účastníky. Tento způsob implementace byl zvolen, protože vytvoření veřejného blockchainu pro jednoho vývozce a více dodavatelů by vyžadovalo více bezpečnostních prvků na ochránění dat v blockchainu a muselo by se zavést připisování pouze samotných hash hodnot do blockchainu. Pro implementaci byla tedy zvolena ta lehčí implementace na úkor ceny.

Povinná účast důvěryhodných entit

Klasické schéma pro bankovní akreditiv a i součást implementace je vytvořena pro 4 účastníky, kteří za standardních podmínek mezi sebou uzavírají dohodu o bankovním akreditivu. Do budoucna by bylo možné díky implementaci skrz blockchain v některých případech i vynechat tyto důvěryhodné autority, které dohlížejí na celý proces akreditivu. Umístěním kontroly do blockchainu by každá strana dostávala aktuální informace o probíhajícím bankovním akreditivu a už by nebylo nutné žádat banky o spoluúčast. Toto vynechání 2 účastníků by pak následně mělo za důsledek snížení ceny samotného procesu. Tato možnost by mohla být realizovaná například u stran, které obchodují navzájem často a už mají vybudovanou nějakou důvěru. Pro nové nebo výjimečné obchody by však byla stále možnost využití implementace pro 4 strany za účelem zvýšení jistoty plnění vzniklých závazků.

10 Závěr

V rámci diplomové práce byl v teoretické části popsán úvod do problematiky blockchain technologie. Popis blockchain technologie začíná od samostatných elementárních bloků až po celek, který je reprezentován jako distribuovaná databázová tabulka. Dále byly popsány různé možnosti dělení blockchain technologie, z kterých jsme následně vybrali ty nejvhodnější pro implementaci na problém správy procesu bankovního akreditivu. V další kapitole týkající se konsenzuálních algoritmů bylo zhodnoceno porovnání těch nejrozšířenějších a to nejen z hlediska bezpečnosti nebo počítačové náročnosti, ale také z hlediska zajištění přístupů k autentičnosti bloků v rámci implementace. Součástí teoretické části této práce je také porovnání a popis aktuálních implementací blockchainu. Do teoretické části je zahrnut i popis a vysvětlení samotného bankovního akreditivu a porovnání s nejčastěji využívaným řešení pro správu bankovního akreditivu. Závěr teoretické části se věnuje náhledu do portfolia SAP a představení toho, jaké služby budeme v rámci implementace využívat pro bankovní akreditiv.

V rámci implementační části práce je popsáno celé nastavení infrastruktury v rámci produktů SAP Cloud Platfrom, které využívají blockchain. Součástí tohoto popisu je uvedení do programovacího jazyka, vysvětlení problematiky napojení jednotlivých součástí, ale také jejich konfigurace. Celková implementace dokáže spravovat proces bankovního akreditivu využívajícího technologii blockchain za dodržení několika podmínek, které jsou zmíněny během implementace. Součástí implementace je jednoduché uživatelské rozhraní pro testování této funkcionality. V přiloženém dokumentu na CD je vysvětlený mechanizmus, jak se s bankovním akreditivem bude pracovat a jaké máme očekávat výstupy. Poslední část implementační části pak shrnuje komplikace, které vznikly během vývoje a jakým způsobem byly nakonec vyřešeny.

Výstupem této práce je bezpečná implementace blockchain technologie v SAP Cloud Platfrom, která inovuje staré řešení pro správu bankovního akreditivu. Implementace zvažuje otázky obchodního aspektu za účelem dosažení co nejlepší kvality služeb ve vztahu k ceně. Implementace byla vytvořena tak, aby bylo jednoduché data z této aplikace následně spravovat nebo exportovat do jiných aplikací v rámci firmy.

Implementace diplomové práce vyřešila několik problému, které vznikly během implementace a jež představují běžné komplikace pro jakýkoliv vývoj aplikací využívajících blockchain. Mezi nejčastější problémy pro implementace blockchainu je GDPR, které omezuje data, která mohou být do blockchainu uložena. Tento problém byl vyřešen tak, že všechna data se budou týkat pouze právnických osob. Navíc implementace využívá privátní blockchain, do něhož budou mít přístupu jen vybraní

účastníci. Problém velké sítě a nutnosti, aby každý jeden účastník měl k dispozici uzel, byl vyřešen na základě přístupu do blockchainu skrz kanál, kde se ověřuje identita při spuštění aplikace. Snížením počtu instancí uzlů decentralizované databáze byla snížena ekonomická náročnost pro provozování této implementace. Jeden z hlavních požadavků bylo zajištění integrity a transparentnosti dat uložených v blockchainu. Tohoto cíle jsme dosáhli právě implementací nejen jedno, ale i záložního uzlu. Navíc každý účastník má k dispozici pouze metody, které sám používá. Nelze tedy předstírat identitu někoho jiné, protože používá metody, které mu byly přiděleny iniciátorem blockchainu sítě. Tímto byl také splněn požadavek na vyšší bezpečnost a kontrolu procesu celého bankovního akreditivu.

Literatura

[1] DRESCHER, Daniel. Blockchain basics. Berkeley, California: Apress, 2017. ISBN 148422603.

[2] GUPTA, Raja, Nagesh CAPARTHY, Vijayalakshmi GOPALAKRISHNA a Atul LADIA. Introducing Blockchain with SAP Leonardo. Boston (MA): Rheinwerk Publishing. ISBN 978-1-4932-1808-0.

[3] RAY, Shaan. Merkle Trees. Hackernoon [online]. USA: Hackernoon, 2017 [cit. 2019-11-16]. Dostupné z: https://hackernoon.com/merkle-trees-181cb4bc30b4

[4] CURRAN, Brian.What is a Merkle Tree? Beginner's Guide to this Blockchain Component. Blockonomi [online]. Palo Alto: Blockonomi, 2019 [cit. 2019-11-16]. Dostupné z: https://blockonomi.com/merkle-tree/

[5] KANSAL, Satwik. Merkle Trees: What They Are and the Problems They Solve. Codementor [online]. Delhi: Codementor, 2018 [cit. 2019-11-16]. Dostupné z: https://www.codementor.io/blog/merkle-trees-5h9arzd3n8

[6] WHAT IS MULTICHAIN TECHNOLOGY?. Blockchain council [online]. San Francisco: Blockchain council, 2017 [cit. 2019-11-16]. Dostupné z: https://www.blockchain-council.org/multichain/multichain-technology/

[7] CBDC Part 5: Comparing Corda, Fabric, and Quorum. Blockchain.news [online]. China: Standard Kepler Research, 2019 [cit. 2019-11-16]. Dostupné z: https://blockchain.news/Post?id=CBDC-Part-5%3A-Comparing-Corda,-Fabric,-and-Quorum-6b2e71a7-7524-46ff-8a77-d35558d029bf

[8] ANWAR, Hasib. Consensus Algorithms: The Root Of The Blockchain Technology. 101Blockchains [online]. New York: 101Blockchains, 2018 [cit. 2019-11-16]. Dostupné z: https://101blockchains.com/consensus-algorithms-blockchain/

[9] FRANKEFIELD, Jake. Proof of Burn (Cryptocurrency). Investopedia [online]. Oberlin: Investopedia, 2018 [cit. 2019-11-16]. Dostupné z: https://www.investopedia.com/terms/p/proof-burn-cryptocurrency.asp

[10] SAINI, Vaibhav. ConsensusPedia: An Encyclopedia of 30+ Consensus Algorithms. Hackernoon [online]. Boston (MA): Hackernoon, 2018 [cit. 2019-11-16]. Dostupné z: https://hackernooncom/consensuspedia-an-encyclopedia-of-29-consensus-algorithms-e9c4b4b7d08f

[11] SAP Cloud Platform. SAP [online]. Germany: SAP, 2019 [cit. 2019-11-17]. Dostupné z: https://www.sap.com/products/cloud-platform.html

[12] LAHL, Dan. SAP Leonardo: Digital Transformation at the Intersection of Technology and Methodology. SAP Leonardo: Digital Transformation at the Intersection of Technology and Methodology [online]. Germany: SAP, 2018 [cit. 2019-11-17]. Dostupné z: https://blogs.sap.com/2018/02/23/sap-leonardo-digital-transformation-at-the-intersection-of-technology-and-methodology/

[13] BRETT, Charles. Blockchain disadvantages: 10 possible reasons not to enthuse. Enterprise times [online]. California: Enterprise times, 2018 [cit. 2020-03-29]. Dostupné z: https://www.enterprisetimes.co.uk/2018/10/15/blockchain-disadvantages-10-possible-reasons-not-to-enthuse/

[14] MUDRAKOLA, Sukesh. BLOCKCHAIN LIMITATIONS: THIS REVOLUTIONARY TECHNOLOGY ISN'T PERFECT — AND HERE'S WHY. TechGenix [online]. Malta: TechGenix, 2018 [cit. 2020-03-29]. Dostupné z: http://techgenix.com/blockchain-limitations/

[15] ŠKORNIČKOVÁ, Eva. Co je GDPR a jak bude aplikováno v Česku. Obecné nařízení o ochraně osobních údajů prakticky [online]. Praha: GDPR.cz, 2019 [cit. 2020-03-29]. Dostupné z: https://www.gdpr.cz/gdpr/co-je-gdpr/

[16] BURDA, Karel. Úvod do kryptografie. Brno: Akademické nakladatelství CERM, 2015. ISBN 978-80-7204-925-7.

[17] BOUŠKA, Petr. Kryptografie a šifrování. Samuraj [online]. Praha: Samuraj, 2019 [cit. 2020-03-29]. Dostupné z: https://www.samuraj-cz.com/clanek/obecny-uvod-do-sifrovani-dat/

[18] What are zk-SNARKs? Zcash [online]. San Francisco: Zcash community, 2019 [cit. 2020-03-29]. Dostupné z: https://z.cash/technology/zksnarks/

[19] DAVIES, Aran. The Ultimate Guide to Blockchain Implementation. DevTeam.Space [online]. Omsk, Rusko: DevTeam.Space, 2019 [cit. 2020-03-29]. Dostupné z: https://www.devteam.space/blog/how-to-integrate-a-blockchain-technology-into-your-project/

[20] MITRA, Rajarshi. Bitcoin VS Ethereum: [The Ultimate Step-by-Step Comparison Guide]. Blockgeeks [online]. Toronto, Kanada: Blockgeeks, 2019 [cit. 2020-03-29]. Dostupné z: https://blockgeeks.com/guides/bitcoin-vs-ethereum-ultimate-comparison-guide/

[21] FERNANDO, Jason. Bitcoin vs. Litecoin: What's the Difference? Investopedia [online]. New York: Investopedia, 2019 [cit. 2020-03-29]. Dostupné z: https://www.investopedia.com/articles/investing/042015/bitcoin-vs-litecoin-whats-difference.asp

[22] Helperbit is the Italian winner of Fintech category. Startup europe awards [online]. Brusel: Startup europe awards, 2018 [cit. 2020-03-29]. Dostupné z: https://startupeuropeawards.eu/helperbit-is-the-italian-winner-of-fintech-category/

[23] KAGAN, Julia. Letter of Credit. Investopedia [online]. New York: Investopedia, 2019 [cit. 2020-03-29]. Dostupné z: https://www.investopedia.com/terms/l/letterofcredit.asp

[24] MT 798. Swift.com [online]. Belgie: Swift.com, 2019 [cit. 2020-03-29]. Dostupné z: https://www.swift.com/our-solutions/corporates/drive-trade-digitisation/mt-798

[25] Letters of Credit - SAP Help Portal. Letters of Credit - SAP Help Portal [online]. Walldorf: SAP, 2019 [cit. 2020-03-29]. Dostupné z: https://help.sap.com/viewer/e5ec5859d8e54df98492d80564a734c0/1911.500/en-US/57dcca5501cf1d22e10000000a44147b.html?q=letter

[26] Validate ID [online]. Barcelona, Španělsko: Validate ID, 2017 [cit. 2020-04-21]. Dostupné z: https://www.validatedid.com/

Seznam symbolů, veličin a zkratek

SWIFT	Společnost pro celosvětovou mezibankovní finanční telekomunikaci – Society for Worldwide Interbank Financial Telecommunication
XML	Rozšiřitelný značkovací jazyk – eXtensible Markup Language
BaaS	Blockchain jako služba – Blockchain as a service
SaaS	Software jako služba – Software as a service
GDPR	Obecné nařízení o ochraně osobních údajů – General Data Protection Regulation
SAP	Systems - Applications - Products in data processing – Systeme, Anwendungen, Produkte in der Datenverarbeitung
ID	Identifikační číslo – Identification data
API	Rozhraní pro programování aplikací– Application programming interface
NIST	Národní institut standardů a technologie – National Institute of Standards and Technology
SWIFT	Společnost pro celosvětovou mezibankovní finanční telekomunikaci – Society for Worldwide Interbank Financial Telecommunication
CPU	Centrální procesorová jednotka – Central processing unit
GPU	Grafický procesor – Graphics processing unit
HANA	Vysoce výkonné analytické zařízení – High-performance analytic appliance
JSON	JavaScriptový objektový zápis – JavaScript Object Notation
UI	Uživatelské rozhraní – User interface
AWS	Amazon webové služby – Amazon web services
CRUD	Vytořit, číst, editovat, smazat – Create, read, update, delete
LC	Bankovní akreditiv – Letter of credit
PaaS	Platforma jako služba – Platform as a service
CD	Kompaktní Disk – Compact Disk
HMAC	Autentizačního kódu zprávy – Keyed-hash Message Authentication Code
MD5	Přehled zpráv 5 – Message-Digest 5
SHA	Rozšířená hašovací funkce – Secure Hash Algorithm
DES	Standard šifrování dat – Data Encryption Standard
AES	Pokročilý standard šifrování – Advanced Encryption Standard
TLS	Zabezpečení transportní vrstvy – Transport Layer Security
RSA	Rivest, Shamir, Adleman šiforvání – Rivest, Shamir, Adleman encryption
zk-SNARK	Nulový-vědomý neinteraktivní argument znalostí – Zero-Knowledge

	Succinct Non-Interactive Argument of Knowledge
XRP	měna Ripple – Ripple
SCP	SAP Cloud Platforma – SAP Cloud Platform

A Obsah přiloženého CD

- `/` ..... kořenový adresář
  - `/src` ..... složka src
    - `LC.go`
    - `LC.yaml`
  - `manual.pdf` ..... Kroky pro sledování průběhu bankovního akreditivu
  - `LC.zip` ..... zip soubor pro verzování
    - `/src`
      - `LC.go`
      - `LC.yaml`
    - `chaincode.yaml`
  - `chaincode.yaml` ..... informace o verzi